

PART II

RESEARCH PAPERS

8. EXPLORATORY ISSUES

- 8.1 FIPER: An Intelligent System for the Optimal Design of Highly Engineered Products
M. W. Bailey, GEAE, USA, and W. H. VerDuin, OAI, USA
- 8.2 Towards an Objective Comparison of Stochastic Optimization Approaches
J. C. Spall, S. D. Hill, and D. R. Stark, The John Hopkins University, USA
- 8.3 Some Measurable Characteristics of Intelligent Computing Systems
C. Landauer, K. Bellman, The Aerospace Corporation, USA
- 8.4 Generalizing Natural Language Representations for Measuring the Intelligence of Systems
A. Meystel, Drexel University, USA
- 8.5 Toward Measures of Intelligence Based On Semiotic Control
C. Joslyn, Los Alamos National Laboratory, USA
- 8.6 Selected Comments on Defining and Measuring of Machine Intelligence
P. K. Davis, RAND, USA
- 8.7 Hierarchic Social Entropy: An Information Theoretic Measure of Robot Group Diversity
T. Balch, Carnegie Mellon University, USA
- 8.8 The Development Approach to Evaluating Artificial Intelligence—A Proposal
A. Treister-Goren, J. Dunietz, AI NV, USA-Israel
- 8.9 General Scientific Premises of Measuring Complex Phenomena
H. M. Hubey, Montclair State University, USA
- 8.10 Using Visualisation for Measuring Intelligence of Constructed Systems
V. Grishin, View Trends, Ltd, USA, and A. Meystel, Drexel University, USA

FIPER: An Intelligent System for the Optimal Design of Highly Engineered Products

Michael W. Bailey, *GE Aircraft Engines, Cincinnati, OH 45215*
and William H. VerDuin, *OAI, Cleveland, OH 44142*

Abstract

This paper outlines the development of an advanced design environment that invokes a new intelligent system paradigm for the design of highly engineered products. The paradigm of the CAD Master Model (MM) is extended with the introduction of the Intelligent Master Model (IMM). The use of knowledge based engineering tools captures *why* and *how* of the design in addition to the *what*.

Turbine engine development is a highly coupled disciplinary process. With ever increasing demands in life cycle costs, environmental aspects (noise, emissions and fuel consumption) and performance, the availability of accurate analytical tools during the design process is a given and ceases to be a discriminator between competitors. The application of these tools and their automated interaction in a robust computational environment may determine the success or failure of a project by reducing design cycle time and avoiding costly rework.

This paper describes pilot projects at GE Aircraft Engines (GEAE) and the productivity metrics that justified broader implementation within GEAE. Developed using the UniGraphics CAD system for the design of aircraft engines, this system is applicable to any highly engineered product. This approach will, with the support of a four year \$21.5M NIST ATP (National Institute of Standards and Technology Advanced Technology Program), be generalized in FIPER (Federated Intelligent Product EnviRonment), a web based environment that will support multi-disciplinary design and optimization.

The Problem

The development of *robust and optimal, highly engineered products and processes* in today's environment of step-function reductions in cycle time, cost take-out, and improved performance seriously tax the capabilities of today's design systems. Further exacerbating the problem is the need to improve and control quality, for both internally manufactured parts and materials and parts produced through supply chains. Since products are now designed, manufactured and serviced at geographically disparate locations, the ability to share relevant product data is critical.

The Solution

FIPER presents a solution in the form of an Integrated Multidisciplinary Design System which

- Exploits the concept of the IMM, permitting context specific views of the MM

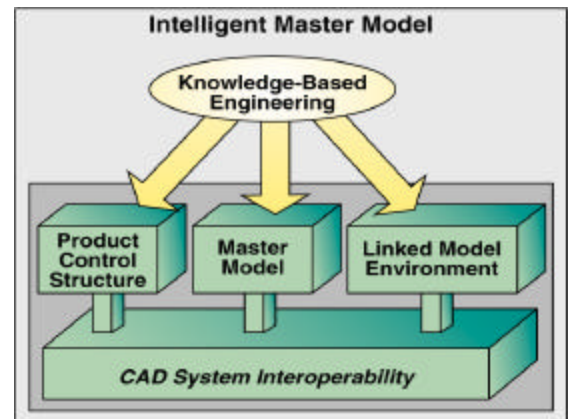
- Seamlessly integrates relevant technologies to enable rapid instantiation and simulation-based evaluation of products and processes

Vision: Integrated Multidisciplinary Design Environment

The integrated multidisciplinary design environment under development will enable users to define process maps and rapidly integrate their own proprietary product-specific design and simulation tools through visual programming techniques. It will automatically provide access to a set of technologies including CAD systems and low and high fidelity analysis modules, as well as Multidisciplinary Optimization (MDO) and Robust Design technologies. It will exploit Knowledge Based Engineering to capture rules and best practices that can drive product definition through the (IMM)

Intelligent Master Model

The Intelligent Master Model (Figure 1) is a major enhancement to the Master Modeling concept. Knowledge Based Engineering (KBE) is fused with Product Control Structure (PCS), conventional MM and Linked Model Environment (LME) to collectively render it an Intelligent Master Model. The IMM captures the intent behind the product design by representing the *why* and *how*, in addition to the *what* of a design. The geometric description is only one view of the information associated with the total product model. The IMM can also contain part dependencies, geometric and non-geometric attributes, manufacturing producibility and cost constraints. IMM can provide access to external databases, and can be integrated with proprietary and commercial codes through the LME.



The IMM can

Figure 1. Intelligent Master Model

capture and archive corporate design practices as well as design and manufacturing engineering expertise. This knowledge can enable less experienced engineers to consistently produce correct first time designs.

The IMM captures the process for generating the PCS at the conceptual and preliminary design level, which then flows the critical information to the detail design and manufacturing. The IMM uses its knowledge base to enable parametric scaling of designs in a top down fashion. When parameters must be computed by execution of simulation codes, the IMM manages this execution by working with process integration tools.

The Master Model

The Master Model captures the requisite information, geometric and non-geometric, to enable context-specific views of necessary design, manufacture, test, and service data. A product design system that supports early requirements definition and flow-down demands that the underlying representation be flexible to geometric, attribute, feature and knowledge-based changes. The traditional CAD representation is flexible only in a geometric sense.

The Master Model (Figure 2) at the lowest or geometric level consists of parametric geometry features such as primitives, extrusions, holes, etc., which form the basic product description. Parameters associated with these geometric features are a subset of the key characteristics which are manipulated to define the product. At this level, the key characteristics include the traditional concepts of dimensionality (length, radius, angle, etc.), as well as those concepts that follow from knowledge-based solid modeling such as offset, spatial alignment, and perpendicularity constraints. Additionally, the existence of a feature is itself an attribute which may be turned on or off as needed to represent the part to varying fidelity levels. For example a bolthole is typically present during a stress analysis but omitted during a computational fluid dynamics analysis. This simplification would be part of the context model, thus creating a context-specific view of the geometry using feature suppression.

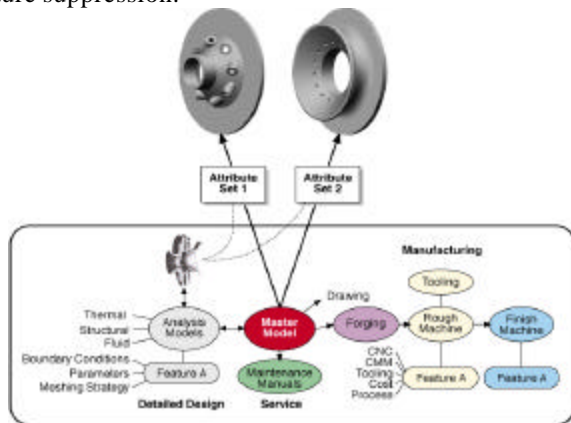


Figure 2. The Master Model supports Feature based Modeling

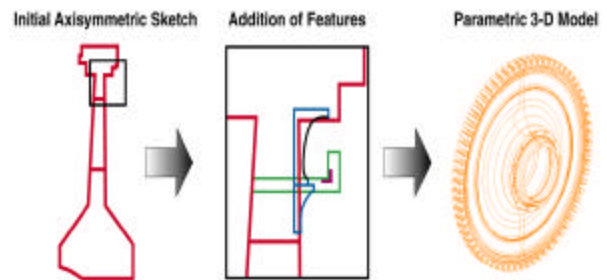


Figure 3. Feature Based Modeling

Using parametric feature-based technology, models are constructed by initially creating simple parametric block shapes to which features (e.g. flanges) are attached. Compound blends are then created and added to the model together with standard features such as radii and chamfers, to create the axisymmetric solid. Finally, non-axisymmetric features such as holes and slots are then added as shown in Figure 3. This feature-based approach is consistent with feature based analytical model building and cost estimating, while also providing feature suppression functionality.

The initial approach to KBE was the encapsulation of product rules within UniGraphics XESS spreadsheets. These spreadsheets are linked to the geometry such that design rules and practices are parameterized to drive geometry. External codes such as those for disk design could also be executed. Thus an increase in flow through the compressor would initiate an aerodynamic resizing of blades and vanes resulting in a blade platform and attachment resizing combined with a disk redesign due to increased centrifugal loads. The whole compressor would thus “rubber band” or parametrically expand to accommodate increased flow.

The Product Control Structure

The PCS facilitates top-down control of the design, allowing the engineer to layout the system configuration and control changes in a top-down fashion. It facilitates *what-if* analysis at the conceptual, preliminary, and detailed design levels by allowing the designer to make parametric changes or to evaluate alternate configurations. This encourages design reuse and enforces standardization in the design process.

The PCS is a hierarchical decomposition of the product into its systems, subsystems and components (Figure 4). These are represented by high-level product attributes and key datum planes and axes to capture their spatial location and orientation. Once the top-level datums have been established and referenced by the subsystems, each subsystem can be designed independently in a distributed manner and later be automatically assembled. Within the PCS, components may be represented by preliminary, simplified geometry (e.g., 2-D cross-sections) or just datums. The cross-sections are picked from a library of cross-section

types based on rules. The values for the parameters that define a cross-section are determined using

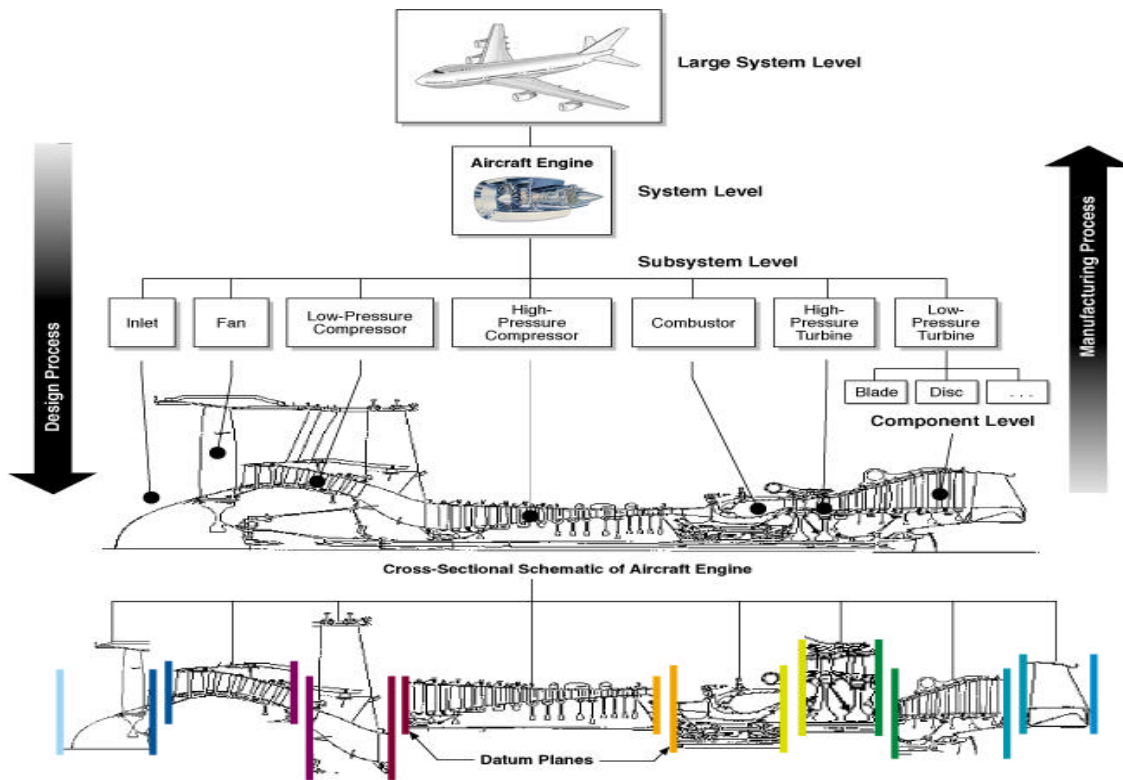


Figure 4. Product Control Structure

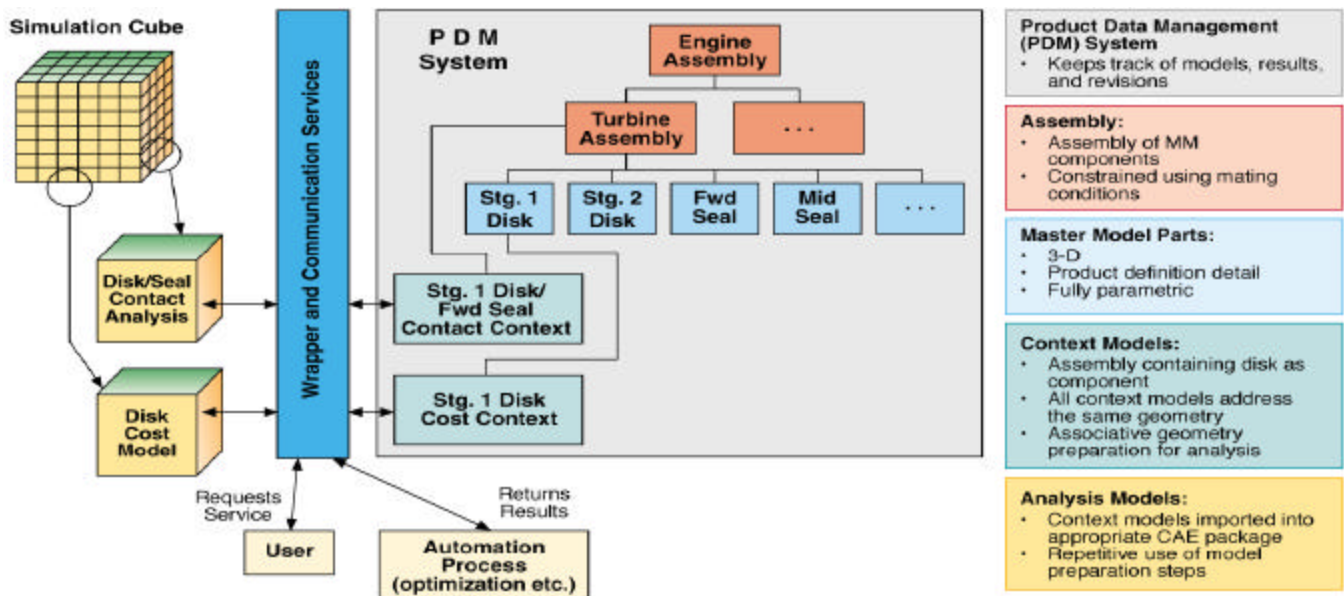


Figure 5. Linked Model Environment

rules captured in the knowledge base. The leaf nodes of the PCS become the seed parts for the bottom-up design of the product into a 3-D assembly. The parts contain 3-D features to capture additional design and manufacturing intent. Everything is fully associative, and thus all changes to the PCS propagate throughout the model.

The Linked Model Environment

Disciplines such as stress analysis, heat transfer analysis, fluids or combustion analysis, and manufacturing and cost

prediction each use their own abstraction of the physical model of the product. Within one discipline, several context-specific views may exist as the design evolves. For example, 2-D axisymmetric stress analysis models and detailed 3-D stress analysis models of various levels of refinement for the individual components of a jet engine are required. Each of these analysis models is associated with one or more simulation tools or codes, from simple response surfaces or performance maps during the conceptual design phase, to more complex analysis codes for detailed design,

manufacturing process simulation, and cost modeling. This provides the promise of geometric zooming. Historically,

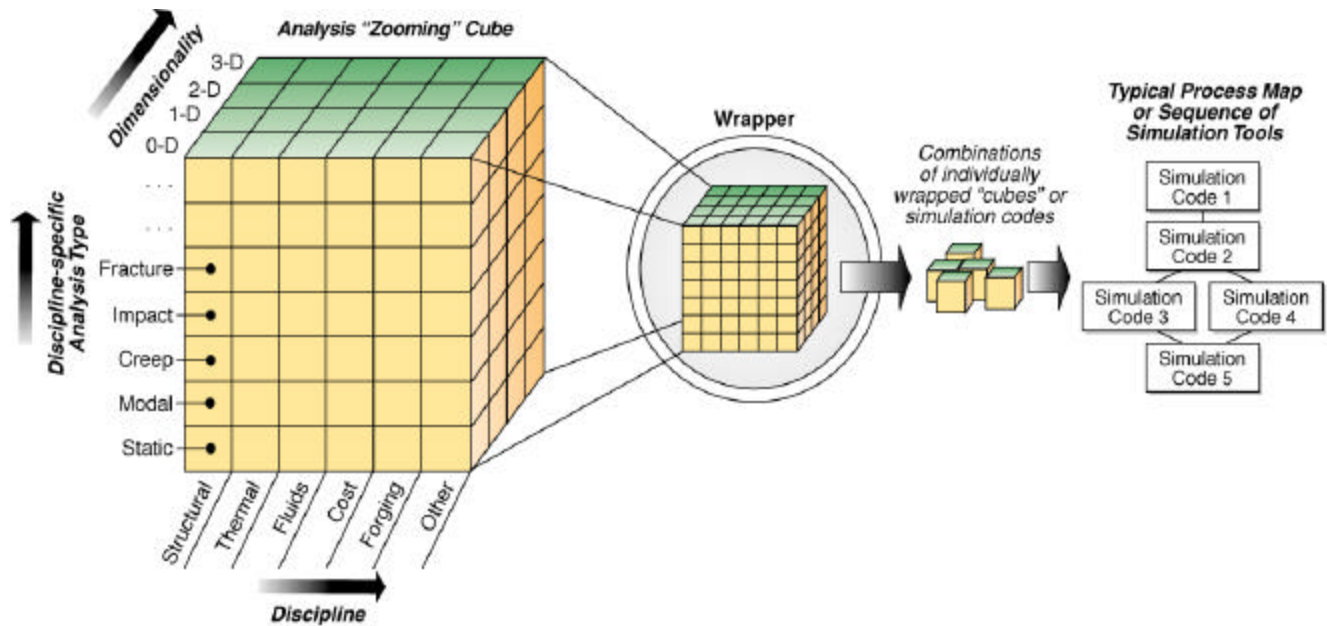


Figure 6. The Simulation Engine

these models exist in a heterogeneous environment, without explicit connections between them. Thus, a design change demanded by one disciplinary group has to be manually incorporated into all the various models of the product that co-exist; a process that is both tedious and error prone. Within the LME (Figure 5) a product's analysis and process models are linked to the Master Model so that all models are automatically synchronized to a single Master Model. Thus, a process is established by which design changes caused by one discipline are fed back to the Master Model. A Product Data Management (PDM) system tracks the design revisions and the associated analysis views or context models of the product.

Simulation Engine

An integral part of the LME is the simulation engine where the analysis tools themselves are wrapped for ease of reuse in a plug-and-pay architecture. To achieve robust and optimal designs, iterative analysis is required. Therefore, ready access to the requisite analysis codes and process maps is essential. The Simulation Engine (Figure 6) provides:

- a programmable mechanism to specify and control the execution of the analysis process
- a mechanism to enable users to easily wrap codes
- an ensemble of pre-wrapped multidisciplinary, variable-fidelity, product-specific analysis tools
- Individual codes and process maps to be linked to the IMM for either manual or automatic execution under program control.

The NASA Glenn Research Center's Numerical Propulsion System Simulator (NPSS) has used a similar cube

representation to show the interconnectivity of functional codes, multiple levels of analysis, and zooming to represent their computer-based engine in a test cell. The Simulation Engine is a generalization of this concept for generic products.

Design For Six Sigma

The goal of Design For Six Sigma (DFSS, 3.4 defects per million opportunities) is to create products and processes which are at Six Sigma levels of performance, manufacturability, reliability and cost. DFSS is based on an orderly process which identifies and flows down Critical to Quality (CTQ) characteristics for the product, process or service. This enables quality measures to be driven into the product during the early design phases where the cost of implementing changes is relatively low in comparison to fixing the problems later in the product life cycle. Key design factors for each CTQ are identified and statistical performance models are developed. Modeling, simulation, Design of Experiments (DoE) and analysis are usually employed to develop the statistical models. The essence of DFSS is to migrate from a deterministic to a probabilistic design approach. DFSS is generally focused on shifting means for CTQ's and reducing variances about means so that customer expectations are met at minimum cost.

Robust Design is an intrinsic part of DFSS. Traditionally optimal design and robust design were viewed as independent technologies, but in fact there is great synergism and common core concepts that can be exploited to achieve

FIPER Role

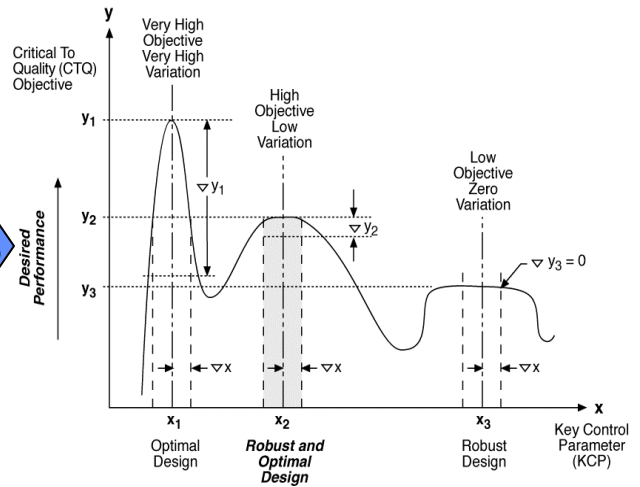
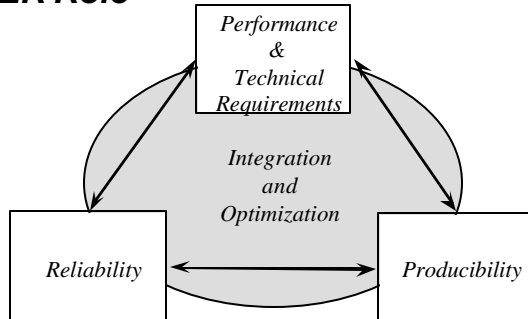


Figure 7. The True Meaning of DFSS

optimal *and* robust designs for products and processes. Optimality and robustness often have competing objectives. The focus of the robust optimization problem is to simultaneously optimize the performance (mean of the response) and minimize the variation. In other words, a maximization problem would not merely strive for the highest peak, but would strive for a high plateau. *In practice this represents a trade-off between Performance and Technical Requirements, Reliability and Producibility* (Figure 6). This represents a paradigm shift in design methodology.

Background

There are many definitions of a Master Model. At GEAE the definition is a single geometric representation, ideally 3-D, created at concept using feature based parametric modeling techniques in a linked associative environment, and utilized through manufacturing. In addition there is an evolution of a tight integration of all elements of a product creation, manufacturing and support permitting true concurrency for analysis and manufacturing since updates can be flowed down to the individual activities from the MM. An additional requirement is the management of all types of data or metadata within the Common Geometry environment. The fusion of a conventional MM with PCS, LME and KBE results in an IMM, the next logical step in CAD's evolution.

Historically analysis codes were coupled together with input and output files; geometry was provided as an output as necessary, probably as an IGES file. The new approach is to have geometry central or *common* to all processes and to use it as a design integrator. This facilitates CAD integration with analysis and manufacturing. Four years ago GE Aircraft

Engines started its Common Geometry initiative, based on UniGraphics and commercial code to the extent possible. The first year focused on strategy. Historically at GEAE conceptual and preliminary design are accomplished using simplifying assumptions in a unique set of tools. Changes in the underlying assumptions and the lack of a rigorous handoff to detail design often meant that the preliminary design was repeated. Since business commitments are made based on preliminary design this increased the risk of meeting customer CTQ requirements. It is well understood that 70 to 80% of a product's cost is locked in during conceptual and preliminary design. Previous efforts had focused on productivity tools that relied heavily on automation. The discovery of UG/WAVE with its top down approach using a Control Structure meant it was possible to drive the design using requirements providing *functional* and *spatial* integration thereby making it possible to create 3-D solid models at the Conceptual/Preliminary Design Phase. This combined with a tight integration of CAD with analysis and manufacturing in LME would provide a truly concurrent design environment.

During the second year three pilots were conducted to demonstrate the technical feasibility and generate metrics for the return on investment analysis necessary to move to a broader implementation across the business. These pilots focused on Conceptual/Preliminary design, Detailed design and Manufacturing. Although these pilots addressed different sections of the engine, success in individual areas would provide confidence to proceed to a broader implementation.

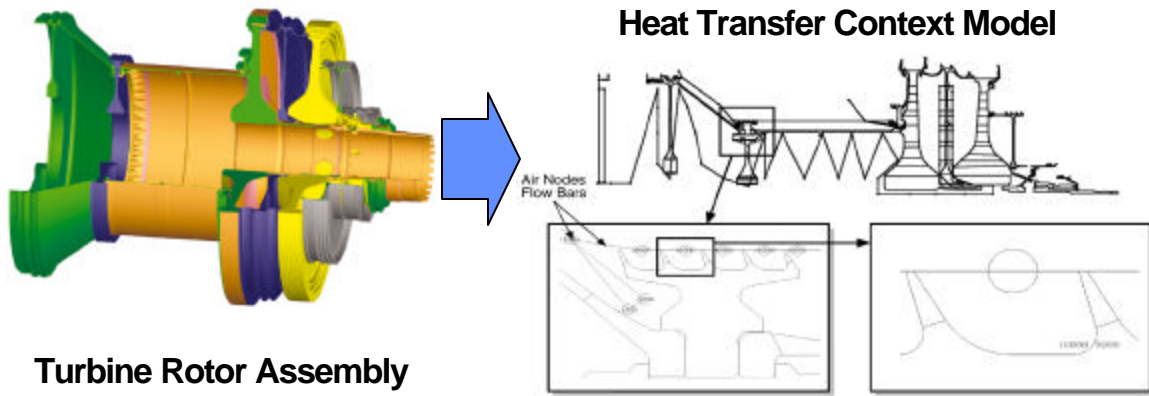


Figure 8. LME Engineering Pilot

SOLID Pilot

The purpose of the SOLID (System Oriented Layout with Integrated Design) pilot was to build a 3-D solid geometry model of a compressor. This was constructed using the UG/WAVE PCS. Model construction is of paramount importance if productivity gains downstream are to be realized. Constructing the model from features enables suppression of selective features by downstream users using context models or “views of geometry”. Traditional 2-D axisymmetric cross sections can still be generated from the 3-D solid. These would be completely associative to the solid and would constitute an output instead of an input. Thus the parameters that drive the 3-D solid would also drive the 2-D cross section. Time invested in constructing the 3-D models facilitates updates as the design evolves. By segregating out the work that would be eliminated using the SOLID model from the charging data from a recently completed program, it was estimated that 34% would be saved at the Conceptual/Preliminary design phase and 7% at the Detailed Design phase.

A key element in the Integration of CAD with Analysis, or any geometry dependent activity, is the creation of context models. A Context model uses the concept of CAD Assemblies to create a “view” of geometry. Just as it is conventional CAD practice to combine parts into assemblies building up into the complete system, it is possible to combine geometry with context information in the form of an assembly. Context in this application means the attachment of information necessary to create a structural, thermal or Computational Fluid Dynamics (CFD) model to geometric entities. The rotor assembly could also be regarded as a context model. This information could be boundary conditions such as pressures, temperatures, loads and the meshing strategy such as mesh seeds or mesh densities. These attributes are applied to the geometric entities in the CAD package.

This context information or “Tagging” should be robust to parametric or non-topological changes and have some robustness to topological changes. A longer term goal is to apply these “Tags” as the analysis model is built in the meshing software, then export these to the CAD software for storage. Currently they are applied in the CAD software. The CAD assembly context model is imported into the meshing software such as PATRAN, ANSYS or ICEM CFD to create the application model. The heat transfer context model is shown in Figure 8. From data accumulated during the pilot, it was estimated that savings of 25% in Detailed Design were possible.

In the manufacturing pilot the focus was using manufacturing context models in conjunction with the 3-D Master Model to generate in process planning and shapes, tooling and Computer Numerically Controlled (CNC) machining tapes. A Low Pressure turbine disk currently in production was used. Note that in the manufacturing environment the modeling works in the opposite sense to detail design. In the detailed design features are added to the model as the design progresses from conceptual through preliminary and detail design; in manufacturing features are removed consistent with manufacturing operations until the raw material remains. Figure 9 shows the associated in-process models and tooling together with the engineering analysis, results and drawing creation. The pilot demonstrated a 15% reduction in process development time and an 80% reduction in process regeneration for parametric changes. In addition the associated tooling was updated when the model was changed.

Computer Measuring Machine (CMM) inspection programs can also be generated from the process models. This is another key context model use of the linked associative environment. Aircraft engine manufacture involves the machining of complex shapes from high temperature alloys that “move” during the manufacturing process. Thus it is important for process control to inspect the process shapes to know what the dimensions are so

adjustments can be made to future machining operations. This offers the possibility of a “real time” machining feedback loop.

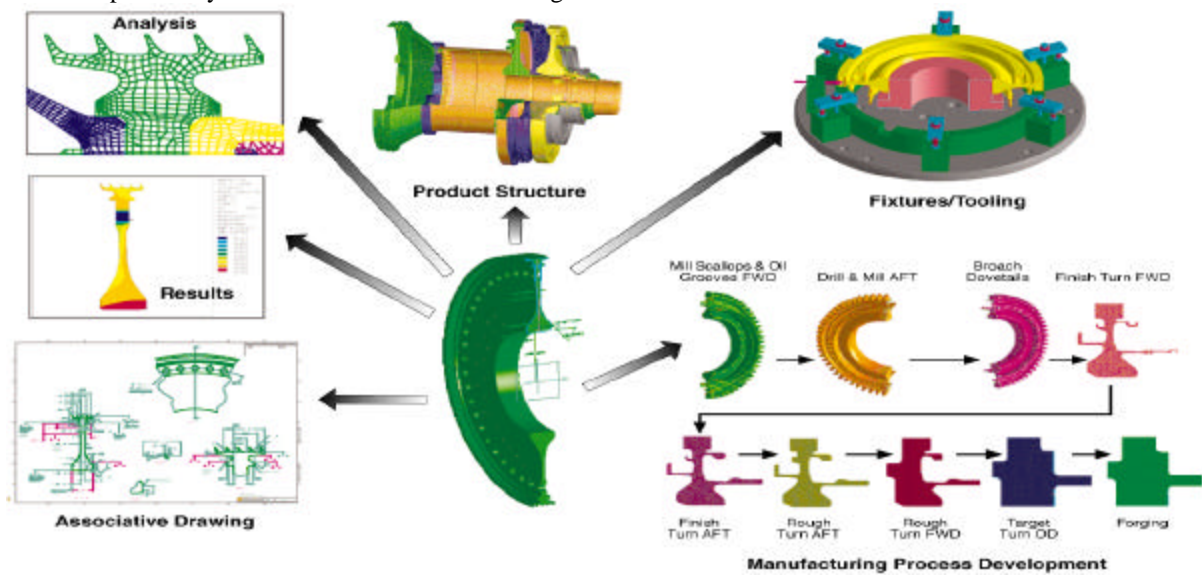


Figure 9. LME Manufacturing Pilot

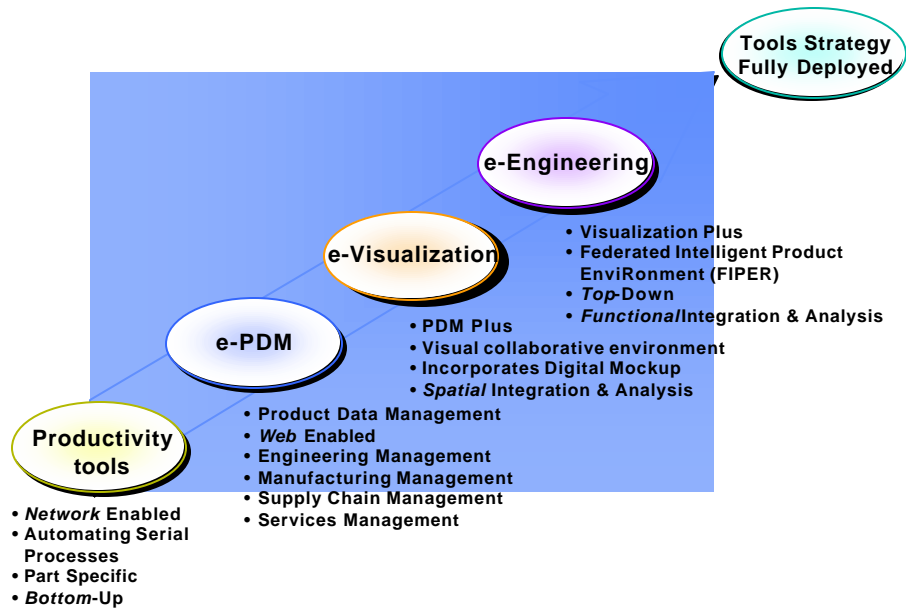


Figure 10. Incremental Approach to Development and Deployment

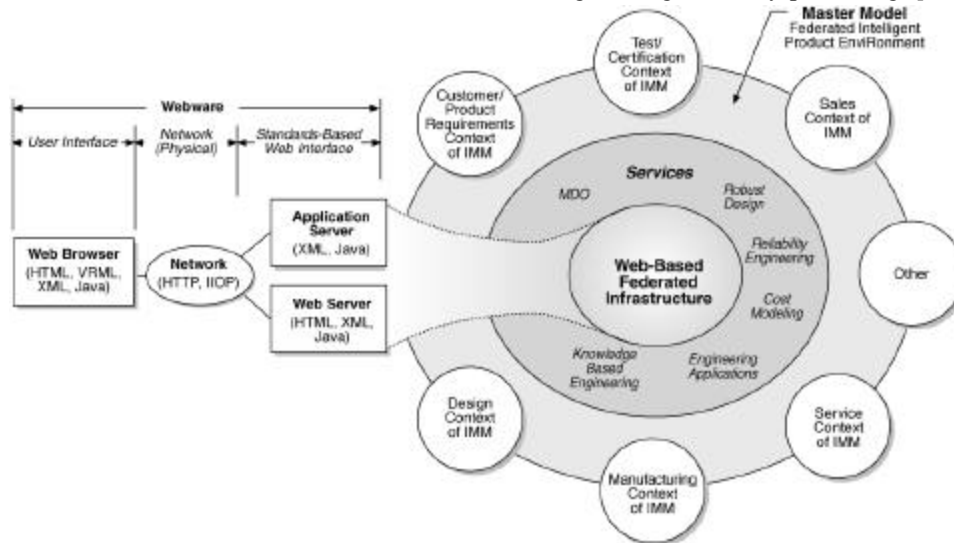
Incremental Approach to Development and Deployment

GEAE's incremental approach to development and deployment is shown in Figure 10.

Productivity Tools/Common Geometry was *network* enabled and automated serial tasks such as mesh creation on individual parts. This can be described as the "run faster" approach and is sub-optimal since it optimizes individual

Management, Manufacturing Management, Supply Chain Management and Services Management with databases, parts lists, process flow, etc. *It would provide the infrastructure for subsequent development.*

e-Visualization represents an enhancement of e-PDM in that it provides a visual collaborative environment incorporating a digital mockup. It thus provides a visual representation of the engineering assembly permitting *spatial* integration and



parts in bottom-up design as opposed to the system design. e-PDM focuses on Product Data Management (PDM) and is *Web* enabled. PDM typically provides Engineering

is with functionality such as interference clearance and removal envelope assessment.

Figure 11. Federated Integrated Design EnviRonnement

e-Engineering builds on the benefits of e-PDM and e-Visualization to provide an environment that supports *functional* integration and analysis providing a top-down or "run smarter" design environment.

The recent FIPER project award by NIST ATP will provide such an environment. Drawing on the experience and qualifications of the FIPER team members and leveraging GE's Corporate commitment to Design For Six Sigma methodology and products, the proposed program will result in the development, demonstration and transition of advanced tools and technology. Key elements of the NIST ATP include:

- Development of an extensible, standards-based plug and play, Web-based architecture to enable the creation of Six Sigma products and processes.
- Development and major enhancement of a set of advanced core technologies necessary to realize Design For Six Sigma, most notably Intelligent Master Modeling, Knowledge Based Engineering, Robust Design, Multidisciplinary Design and Optimization, Cost Modeling and Producibility.
- Demonstration of FIPER on a diverse set of demanding applications, which span conceptual design, through

manufacturing for systems, subsystems and components.

- Dissemination of the technology through a well founded commercialization plan, complimentary teaming, Web-based access, publications, educational programs and the creation of an early adoption program.

Thus FIPER represents a paradigm shift for product development through the introduction of a standards based product development environment Conceptually the FIPER environment is described in Figure 11 and in more detail in Reference1.

The team was chosen for their complimentary roles in achieving the overall FIPER objectives. GEAE is a complex engineering system developer and manufacturer and a Unigraphics CAD system user. Parker Hannifin is a complex aircraft engine and aircraft subsystem and component supplier and a ProEngineer CAD system user. BFGoodrichAerospace is a complex aircraft sub-system and component supplier and CATIA CAD system user. Thus with CAD interoperability being one of the major FIPER initiatives, three out of the four major CAD systems is represented. The fourth, SDRC IDEAS Master Series will be addressed at a later stage, possibly through the early

adopter program. GE Corporate Research and Development (CR&D) has been developing the technology associated with IMM, KBE, MDO and DFSS for a number of years. Engineous Software Inc. is the commercializer for the FIPER software and their current product is iSIGHT, an engineering analysis process integration and optimization tool. Ohio University is providing computer system integration software wrapping tools and is developing a cost model that will be integrated with the IMM. Stanford University is creating producibility models that will be integrated with the IMM. OAI (Ohio Aerospace Institute) is the sponsoring organization and provides program administration. The complimentary teaming are key to the technical and commercial success of the FIPER project.

Acknowledgements

This research is jointly funded through the National Institute for Standards and Technology Advanced Technology Program (NIST ATP) and the General Electric Company. The authors would like to acknowledge this support as well as the valuable input from the whole FIPER team.

References

Ref. 1: Röhl, P. J., Kolonay, R. M. et al. *A Federated Intelligent Product Environment* AIAA-2000-4902, 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, September 6-8, 2000

Towards an Objective Comparison of Stochastic Optimization Approaches

James C. Spall, Stacy D. Hill, and David R. Stark

The Johns Hopkins University
Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, Maryland 20723-6099 U.S.A.

ABSTRACT

This paper is a first step to formal comparisons of several leading optimization algorithms, establishing guidance to practitioners for when to use or not use a particular method. The focus in this paper is four general algorithm forms: random search, simultaneous perturbation stochastic approximation, simulated annealing, and evolutionary computation. We summarize the available theoretical results on rates of convergence for the four algorithm forms and then use the theoretical results to draw some preliminary conclusions on the relative efficiency. Our aim is to sort out some of the competing claims of efficiency and to suggest a structure for comparison that is more general and transferable than the usual problem-specific numerical studies. Work remains to be done to generalize and extend the results to problems and algorithms of the type frequently seen in practice.

KEYWORDS: *Rate of convergence; random search; simultaneous perturbation stochastic approximation; simulated annealing; evolutionary computation.*

1. INTRODUCTION

To address the shortcomings of classical deterministic algorithms, a number of powerful optimization algorithms with embedded randomness have been developed. The population-based methods of evolutionary computation are only one class among many of these available *stochastic* optimization algorithms. Hence, a user facing a challenging optimization problem for which a stochastic optimization method is appropriate meets the daunting task of determining which algorithm is appropriate for a given problem. This choice is made more difficult by the large amount of “hype” and dubious claims that are associated with some popular

algorithms. An inappropriate approach may lead to a large waste of resources, both from the view of wasted efforts in implementation and from the view of the resulting suboptimal solution to the optimization problem of interest.

Hence, there is a need for objective analysis of the relative merits and shortcomings of leading approaches to stochastic optimization. This need has certainly been recognized by others, as illustrated in the recent 1998 IEEE International Conference on Evolutionary Computation, where one of the major subject divisions in the conference was devoted to comparing algorithms. Nevertheless, virtually all comparisons have been numerical tests on specific problems. Although sometimes enlightening, such comparisons are severely limited in the *general* insight they provide. On the other end of the spectrum are the “No Free Lunch Theorems” (Wolpert and McReady, 1997), which simultaneously considers all possible loss functions and thereby draw conclusions that have limited practical utility since one always has at least *some* knowledge of the nature of the loss function being minimized.

Our aim in this paper is to lay a framework for a *theoretical* comparison of efficiency applicable to a broad class of practical problems where some (incomplete) knowledge is available about the nature of the loss function. We will consider four basic algorithm forms—random search, simultaneous perturbation stochastic approximation (SPSA), simulated annealing, and evolutionary computation via genetic algorithms—in the context of continuous variable optimization. The basic optimization problem corresponds to finding an optimal point θ^* :

$$\theta^* = \arg \min_{\theta \in D} L(\theta),$$

* A more complete version of this manuscript is available upon request (james.spall@jhuapl.edu). This work was partially supported by the JHU/APL IRAD Program and U.S. Navy contract N00024-98-D-8124.

where $L(\theta)$ is the loss function to be minimized, D is the domain over which the search will occur, and θ is a p -dimensional (say) vector of parameters. We are mainly interested in the typical case where θ^* is a *unique* global minimum.

Although many stochastic optimization algorithms other than the four above exist, we are restricting ourselves to the four general forms in order to be able to make tangible progress (note that there are various specific implementations of each of these general algorithm forms). These four algorithms are general-purpose optimizers with powerful capabilities for serious multivariate optimization problems. Further, they have in common the requirement that they only need measurements of the objective function, not requiring the gradient or Hessian of the loss function.

2. NO FREE LUNCH THEOREMS AND THEIR RELATIONSHIP TO RATE OF CONVERGENCE

Wolpert and Macready (1997) present a formal analysis of search algorithms for optimization, the most popular of which are evolutionary computation, simulated annealing (SAN) and random search. This work results in several “No Free Lunch Theorems,” stating, in essence, that no algorithm is universally better than other algorithms. The full version of this paper goes into some detail on the implications of these theorems.

3. SIMPLE GLOBAL RANDOM SEARCH

We first establish a rate of convergence result for the simplest random search method where we repeatedly sample over the domain of interest, $D \subseteq R^p$. This can be done in recursive form or in “batch” (non-recursive) form by simply laying down a number of points in D and taking as our estimate of θ^* that value of θ yielding the lowest L value. It is well known that the random search algorithm above will converge in some stochastic sense under modest conditions (e.g., Solis and Wets, 1981; Spall, 2000b):

To evaluate the *rate* of convergence, let us specify a “satisfactory region” $S(\theta^*)$ representing some neighborhood of θ^* providing acceptable accuracy in our solution (e.g., $S(\theta^*)$ might represent a hypercube about θ^* with the length of each side representing a tolerable error in each coordinate of θ). An expression related to the rate of convergence of Algorithm A is then given by

$$P(\hat{\theta}_k \in S(\theta^*)) = 1 - [1 - P(\theta_{\text{new}}(k) \in S(\theta^*))]^k \quad (3.1)$$

We will use this expression in Section 7 to derive a convenient formula for comparison of efficiency with other algorithms.

4. SIMULTANEOUS PERTURBATION STOCHASTIC APPROXIMATION

The next algorithm we consider is SPSA. This algorithm is designed for continuous variable optimization problems. Unlike the other algorithms here, SPSA is fundamentally oriented to the case of *noisy* function measurements and most of the theory is in that framework. This will make for a difficult comparison with the other algorithms, but Section 7 will attempt a comparison nonetheless. The SPSA algorithm works by iterating from an initial guess of the optimal θ , where the iteration process depends on a highly efficient “simultaneous perturbation” approximation to the gradient $g(\theta) \equiv \partial L(\theta)/\partial \theta$.

The SPSA procedure is in the general recursive SA form:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) \quad (4.1)$$

where $\hat{g}_k(\hat{\theta}_k)$ is the SP estimate of the gradient $g(\theta) \equiv \partial L/\partial \theta$ at the iterate $\hat{\theta}_k$ (Spall, 1992) based on the measurements of the loss function and $a_k > 0$ is a “gain” sequence. This iterate can be shown to converge under reasonable conditions (e.g., Spall, 1992; Dippon and Renz, 1997). The essential basis for efficiency of SPSA in multivariate problems is due to the gradient approximation, which uses only two measurements of the loss function to estimate the p -dimensional gradient vector for any p . This contrasts with the standard finite difference method of gradient approximation, which requires $2p$ measurements.

Most relevant to the comparative analysis goals of this paper is the asymptotic distribution of the iterate. This was derived in Spall (1992), with further developments in Chin (1997), Dippon and Renz (1997), and Spall (2000a). Essentially, it is known that under appropriate conditions,

$$k^{\beta/2}(\hat{\theta}_k - \theta^*) \xrightarrow{\text{dist}} N(\mu, \Sigma) \text{ as } k \rightarrow \infty, \quad (4.2)$$

where $\beta > 0$ depends on the choice of gain sequences (a_k and c_k), μ depends on both the Hessian and the third derivatives of $L(\theta)$ at θ^* (note that in general, $\mu \neq 0$ in contrast to many well-known asymptotic normality results in estimation), and Σ depends on the Hessian matrix at θ^* and the variance of the noise in the loss measurements. Unfortunately, (4.2) is not directly usable in our comparative studies here since the other three algorithms being considered here appear to have

convergence rate results only for the case of *noise-free* loss measurements. Recent results by Gerencsér (1999) and Gerencsér and Vágó (2000) on noise-free SPSA may ultimately be useful.

5. SIMULATED ANNEALING ALGORITHMS

The simulated annealing (SAN) method (Metropolis et al., 1953; Kirkpatrick et al., 1983) was originally developed for optimization over finite sets. The Metropolis method produces a sequence that converges in probability to the set of global minima of the loss function as T_k , the *temperature*, converges to zero. Geman and Hwang (1986) present a SAN algorithm for continuous parameter optimization. Their algorithm produces a *continuous-time* stochastic process—a diffusion process—whose probability distributions converge weakly to the uniform probability distribution concentrated on the (global) minima of the loss function, as the temperature decreases to zero.

More recently, Gelfand and Mitter (1993) obtained discrete-time recursions for Metropolis-type SAN algorithms that, in the limit, optimize continuous parameter loss functions: Suppose that $\{\hat{\theta}_k\}$ is a Metropolis SAN sequence for optimizing L and assume that the gradient g of L exists (it does not have to be actually computed).

Furthermore, like SPSA, SAN has an asymptotic normality result (but unlike SPSA, this result applies in the noise-free case). Let $H(\theta^*)$ denote the Hessian of $L(\theta)$ evaluated at θ^* and let I_p denote the $p \times p$ identity matrix. Yin (1999) showed that for $b_k = (b/(k^\gamma \log(k^{1-\gamma} + B_0)))^{1/2}$,

$$[\log(k^{1-\gamma} + B_0)]^{1/2}(\hat{\theta}_k - \theta^*) \xrightarrow{d} N(0, S) \text{ in distribution,}$$

where $\Sigma H + H^T \Sigma + (b/a)I = 0$.

6. EVOLUTIONARY COMPUTATION

There are three general approaches in evolutionary computation, namely Evolutionary Programming (EP), Evolutionary Strategies (ES) and Genetic Algorithms (GA). All three approaches work with a population of candidate solutions and randomly alter the solutions over a sequence of generations according to evolutionary operations of competitive selection, mutation and sometimes

recombination (reproduction). The fitness of each population element to survive into the next generation is determined by a selection scheme based on evaluating the loss function for each element of the population. The selection scheme is such that the most favorable elements of the population tend to survive into the next generation while the unfavorable elements tend to perish.

The principle differences in the three approaches are the selection of evolutionary operators used to perform the search and the computer representation of the candidate solutions. EP uses selection and mutation only to generate new solutions. While both ES and GA use selection, recombination and mutation, recombination is used more extensively in GA. A GA traditionally performs evolutionary operations using binary encoding of the solution space, while EP and ES perform the operations using real-coded solutions. The GA also has a real-coded form and there is some indication that the real-coded GA may be more efficient and provide greater precision than the binary-coded GA. The distinction among the three approaches has begun to blur as new hybrid versions of EC algorithms have arisen.

Global convergence results can be given for a broad class of problems, but the same can not be said for convergence *rates*. The most practically useful convergence rates for EC algorithms seem to be for the class of strongly convex fitness functions. The following result due to Rudolph (1997b) is an extension of a more general result by Rappl (1989). The theorem will be the starting place for the specific convergence rate result that will be used for comparison in Section 7. A more complete discussion of the relevant EC theory is in the full version of the paper.

An EC algorithm has a *geometric rate of convergence* if and only if $E[L_k^* - L(\theta^*)] = O(c^k)$ where $c \in (0, 1)$ is called the *convergence rate*. Under conditions, the convergence rate result for a $(1, \lambda)$ -ES using selection and mutation only on a strongly convex fitness function is geometric with a rate of convergence

$$c = (1 - M_{\lambda,p}^2/Q^2) \text{ where } M_{\lambda,p} = E[B_{\lambda,\lambda}] > 0$$

and where $B_{\lambda,\lambda}$ denotes the maximum of λ independent identically distributed Beta random variables. The computation of $M_{\lambda,p}$ is apparently very complicated since it depends on both the number of offspring λ and the problem dimension p . An asymptotic approximation for the convergence rate for the (N, λ) -ES where offspring are only obtained by mutation is $c \leq [1 - (2p^{-1} \log(\lambda N))/Q^2]$.

7. COMPARATIVE ANALYSIS

7.1 Problem Statement and Summary of Efficiency Theory for the Four Algorithms

This section uses the specific algorithm results in Sections 3 to 6 above in drawing conclusions on the relative performance of the four algorithms. There are obviously many ways one can express the rate of convergence, but it is expected that, to the extent they are based on the theory outlined above, the various ways will lead to broadly similar conclusions. We will address the rate of convergence by focusing on the question:

With some high probability $1 - \rho$ (ρ a small number), how many $L(\theta)$ function evaluations, say n , are needed to achieve a solution lying in some “satisfactory set” $S(\theta^)$ containing θ^* ?*

For each of the four algorithms, we will outline below an analytical expression useful in addressing the question. After we have discussed the analytical expressions, we present a comparative analysis in a simple problem setting for varying p .

Random Search

We can use (3.1) to answer the question above. Setting the left-hand side of (3.2) to $1 - \rho$ and supposing that there is a constant sampling probability $P^* = P(\theta_{\text{new}}(k) \in S(\theta^*)) \forall k$, we have

$$n = \frac{\log \rho}{\log(1 - P^*)}. \quad (7.1)$$

Simultaneous Perturbation Stochastic Approximation

From the fact that SPSA uses two $L(\theta^*)$ evaluations per iteration, the value n to achieve the desired probability for $\hat{\theta}_k \in S(\theta^*)$ is then

$$n = 2 \left(\frac{2d(p)\sigma}{\delta s} \right)^3.$$

Simulated Annealing

The value n to achieve the desired probability for $\hat{\theta}_k \in S(\theta^*)$ is

$$\log n^{1-\gamma} = \left(\frac{2d(p)\sigma}{\delta s} \right)^2.$$

Evolutionary Strategy

The full version of the paper employs Markov's inequality and the bound in Rudolph (1997b) to show that for each generation k , there are λ evaluations of the fitness function so that $n = \lambda k$, where

$$k = \frac{\log \rho - \log(1/\epsilon)}{\log \left[1 - \frac{2}{pQ^2} \log(\lambda/N) \right]}.$$

7.2 Application of Convergence Rate Expressions for Varying p

We now apply the results above to demonstrate relative efficiency for varying p . Let $D = [0, 1]^p$ (the p -dimensional hypercube with minimum and maximum θ values of 0 and 1 for each component). We want to guarantee with probability 0.90 that each element of θ is within 0.04 units of the optimal. Let the (unknown) true θ, θ^* , lie in $(0.04, 0.96)^p$. The individual components of θ^* are θ_i^* . Hence,

$$S(\theta^*) = [\theta_1^* - 0.04, \theta_1^* + 0.04] \times [\theta_2^* - 0.04, \theta_2^* + 0.04] \times \dots \times [\theta_p^* - 0.04, \theta_p^* + 0.04] \subset D.$$

Table 7.1 is a summary of relative efficiency for the setting above for $p = 2, 5$, and 10; the efficiency was normalized so that all algorithms performed equally at $p = 1$, as described below. The numbers in Table 7.1 are the ratios of the number of loss measurements for the given algorithm over the number for the best algorithm at the specified p ; the highlighted values 1.0 indicate the best algorithm for each of the values of p . To establish a fair basis for comparison, we fixed the various parameters in the expressions above (e.g., σ in SPSA and SAN, ρ for the ES, etc.) so that the algorithms produced identical efficiency results for $p = 1$.

Table 7.1. Ratios of loss measurements needed relative to best algorithm at each p for $1 \leq p \leq 10$

	$p = 1$	$p = 2$	$p = 5$	$p = 10$
<i>Rand. Search</i>	<u>1.0</u>	11.6	8970	2.0×10^9
<i>SPSA</i>	<u>1.0</u>	1.5	<u>1.0</u>	<u>1.0</u>
<i>SAN</i>	<u>1.0</u>	<u>1.0</u>	2.2	4.1
<i>ES</i>	<u>1.0</u>	1.9	1.9	2.8

Table 7.1 illustrates the explosive growth in the relative (and absolute) number of loss evaluations needed as p increases for the random search algorithm. The other algorithms perform more comparably, but there are still some non-negligible differences. For example, at $p = 5$, SAN will take 2.2 times more loss measurements than SPSA to achieve the objective of having $\hat{\theta}_k$ inside $S(\theta^*)$ with probability 0.90. Of course, as p increases, all algorithms take more measurements; the table only shows relative numbers of function evaluations (considered more reliable than absolute numbers).

This large improvement of SPSA and SAN relative to random search may partly result from the more restrictive regularity conditions of SPSA and SAN (i.e., for formal convergence, SPSA assumes a unimodal, several-times-differentiable loss function) and partly from the fact that SPSA and SAN work with *implicit* gradient information via gradient approximations. The performance for ES is quite good. The restriction to strongly convex fitness functions, however, gives the ES in this setting a strong structure not available to the other algorithms. It remains unclear what practical theoretical conclusions can be drawn on a broader class of problems.

REFERENCES

- Bäck, T., Hoffmeister, F., and Schwefel, H.-P. (1991), “A Survey of Evolution Strategies,” in *Proceedings of the Fourth International Conference on Genetic Algorithms* (R.K. Belew and L.B. Booker, eds.), pp. 2-9.
- Beyer, H.-G. (1995), “Toward a Theory of Evolution Strategies: On the Benefits of Sex—the $(\mu/\mu, \lambda)$ Theory,” *Evolutionary Computation*, vol. 3, pp. 81-111.
- Chin, D.C. (1994), “A More Efficient Global Optimization Algorithm Based on Styblinski and Tang,” *Neural Networks*, vol. 7, pp. 573-574.
- Chin, D.C. (1997), “Comparative Study of Stochastic Algorithms for System Optimization Based On Gradient Approximations,” *IEEE Transactions on Systems, Man, and Cybernetics—B*, vol. 27, pp. 244-249.
- Culberson, J.C. (1998), “On the Futility of Blind Search: An Algorithmic View of ‘No Free Lunch’,” *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 109-127.
- Dippon, J. and Renz, J. (1997), “Weighted Means in Stochastic Approximation of Minima,” *SIAM Journal on Control and Optimization*, vol. 35, pp. 1811-1827.
- Fabian, V. (1968), “On Asymptotic Normality in Stochastic Approximation,” *Annals of Mathematical Statistics*, vol. 39, pp. 1327-1332.
- Gelfand, S. and Mitter, S.K. (1993), “Metropolis-Type Annealing Algorithms for Global Optimization in R^d ,” *SIAM Journal of Control and Optimization*, vol. 31, pp. 111-131.
- Geman, S. and Hwang, C.-R. (1986), “Diffusions for Global Optimization,” *SIAM Journal of Control and Optimization*, vol. 24, pp. 1031-1043.
- Gerencsér, L. (1999), “Convergence Rate of Moments in Stochastic Approximation with Simultaneous Perturbation Gradient Approximation and Resetting,” *IEEE Transactions on Automatic Control*, vol. 44, pp. 894-905.
- Gerencsér, L. and Vágó, Z. (2000), “SPSA in Noise-Free Optimization,” in *Proceedings of the American Control Conference*, pp. 3284-3288.
- Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983), “Optimization by Simulated Annealing,” *Science*, vol. 220, pp. 671-680.
- Maryak, J.L. and Chin, D.C. (2000), “Stochastic Approximation for Global Random Optimization,” in *Proceedings of the American Control Conference*, pp. 3294-3298.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M. Teller, A. and Teller, E. (1953), “Equation of State Calculations by Fast Computing Machines,” *Journal of Chemical Physics*, vol. 21, pp. 1087-1092.
- Nemirovsky, A.S. and Yudin, D.B. (1983), *Problem Complexity and Method Efficiency in Optimization*, Wiley, Chichester.
- Rappl, G. (1989), “On Linear Convergence of a Class of Random Search Algorithms,” *Zeitschrift für angewandte Mathematik und Mechanik (ZAMM)*, vol. 69, pp. 37-45.

- Rudolph, G. (1994), "Convergence Analysis of Canonical Genetic Algorithms", *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 96-101.
- Rudolph, G. (1996), "Convergence of Evolutionary Algorithms in General Search Spaces," in *Proceedings of the Third IEEE Conference on Evolutionary Computation*, pp. 50-54.
- Rudolph, G. (1997a), *Convergence Properties of Evolutionary Algorithms*, Kovac, Hamburg
- Rudolph, G. (1997b), "Convergence Rates of Evolutionary Algorithms for a Class of Convex Objective Functions," *Control and Cybernetics*, vol. 26, pp. 375-390.
- Rudolph, G. (1998), "Finite Markov Chain Results in Evolutionary Computation: A Tour d'Horizon," *Fundamenta Informaticae*, vol. 34, pp. 1-22.
- Solis, F.J. and Wets, J.B. (1981), "Minimization by Random Search Techniques," *Mathematics of Operations Research*, vol. 6, pp. 19-30.
- Spall, J.C. (1992), "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation," *IEEE Transactions on Automatic Control*, vol. 37, pp. 332-341.
- Spall, J.C. (2000a), "Adaptive Stochastic Approximation by the Simultaneous Perturbation Method," *IEEE Transactions on Automatic Control*, vol. 45, in press.
- Spall, J.C. (2000b), *Introduction to Stochastic Search and Optimization*, Wiley, New York, in preparation.
- Tong, Y.L. (1980), *Probability Inequalities in Multivariate Distributions*, Academic, New York.
- Wolpert, D.H. and Macready, W.G. (1997), "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 67-82.
- Yin, G.G. (1999), "Rates of Convergence for a Class of Global Stochastic Optimization Algorithms," *SIAM Journal on Optimization*, vol. 10, pp. 99-120.

Some Measurable Characteristics of Intelligent Computing Systems

Christopher Landauer, Kirstie Bellman
Aerospace Integration Science Center
The Aerospace Corporation, Mail Stop M6/214
P. O. Box 92957, Los Angeles, California 90009-2957, USA
cal@aero.org, bellman@aero.org

Abstract

We discuss the following measurable characteristics of intelligent behavior in computing systems: (1) speed and scope of adaptability to unforeseen situations, including recognition, assessment, proposals, selection, and execution; (2) rate of effective learning of observations, behavior patterns, facts, tools, methods, etc., which requires identification, encapsulation, and recall; (3) accurate modeling and prediction of the relevant external environment, which includes the ability to make more effective abstractions; (4) speed and clarity of problem identification and formulation; (5) effective association and evaluation of disparate information; (6) identification of more important assumptions and prerequisites; (7) use of symbolic language, including the range and use of analogies and metaphors (this is about identification of similarities), and the invention of symbolic language, which includes creating effective notations. We make no claim that these are all the important characteristics; discovering others is the point of our research program.

Key Phrases: *Intelligent Autonomous Systems, Measuring Intelligent Behavior, Constructed Complex Systems, Reflective Infrastructure*

1. Introduction

This paper will describe some characteristics of intelligent computing systems, describe how to make measurements of those characteristics, and discuss what they might mean, though we know that they do not cover the full spectrum of what is commonly considered to be intelligent behavior. We extract these measurements from several different viewpoints about what is important for intelligent behavior, and explain their most popularly expected implications.

Intelligence is difficult to measure, because it is thought to be an intrinsic property of systems, like a potential capability or competence, whereas the only things that can be measured are actual performances under various kinds of conditions. This problem has plagued the evaluators of human intelligence since the beginning, to the point that they have generally concentrated on measuring some postulated corresponding

performance characteristics [8].

Therefore, metrics can only be based on observed system behavior (though the observations can, of course, measure internal processes from an internal perspective, since we can have some kinds of internal access), since we have no direct access to how internal organization and structure affect intelligence. Even if we assume that intelligence is entirely intrinsic, we cannot evaluate it separately from its corresponding behavior (even if the behavior is only observable introspectively). Measuring performance to infer competence, even of externally observable behavior, is also very difficult and time-consuming, since we intend to use the measurements over a range of situations in order to evaluate the intelligence of different systems.

Success in a particular task is not by itself the right criterion (even if success were well-defined). Many intelligent decisions founder on the rocks of poor information and / or unexpected events, and brute force can make up for a lack of intelligence (e.g., Deep Blue's defeat of Kasparov relied on very fast special-purpose hardware).

Computer programs that play combinatorial games or search the web are not very interesting to us from an intelligent systems point of view, because their domain is so limited and their goals are provided from the outside. Even so, we're interested in computer programs as creative entities (co-investigators, so to speak, instead of just tools), and we think that a careful study of what we can make programs do will be helpful in understanding what the issues are [2] [4]. In order to study these possibilities, we want to define a set of measurements that can be used to differentiate and understand the relationships among different kinds of behavioral characteristics.

We consider autonomy to be more than choosing methods to satisfy goals. A system is autonomous to the extent that it also chooses those goals. In fact, there are really only two classes of (difficult) requirements for effective autonomy: robustness and timeliness. Robustness means graceful degradation in increasingly hostile environments, which to us *implies* a requirement for adaptability, and timeliness means that situations are recognized "well enough" and "soon enough", and that "good enough" actions are taken "soon enough". There is

almost never any *optimization* here (that almost always takes too much time and requires too much information).

For the purposes of this paper, we concentrate on the measurement problem instead of the construction problem, though we have some definite ideas about how to build these interesting programs, based on our Wrapping infrastructure for Constructed Complex Systems [17] [21] [22].

2. System Behaviors

We'll start with the assumption that a computing system is designed to help its users *do* something [9]. That something is a problem in some subject area, such as, for example, copy a file in a computer system, produce a document in a legal office, kill monsters and collect treasures in a computer game, retrieve a web page for a user, solve an equation in a mathematical subject area, find patterns in noisy data in a scientific field, coordinate a distributed simulation for a military application, launch a spacecraft in the aerospace business, collaborate remotely on a design problem for space systems, etc.. We'll use these cases as illustrations in the rest of the discussion.

In all of these cases, there is an *application domain*, which provides a certain context of use and corresponding terminology. Actually, this is more of a *domain-specific language*, since it includes more than just vocabulary terms. It also has a set of abbreviations and conventions about what can remain implicit, and a set of simplifications (which are fruitful lies about the entities and behaviors in the domain). It is important to note that these languages might or might not be written symbolically, since, for example, a computer game is often commanded using a joystick instead of typed commands, and some immersive Virtual Environments are commanded by user movement and gesture.

What the user wants to do is called the *problem*, which only makes sense within the context of interpretation provided by the domain-specific language of the application domain. These languages are used to define the problem context or *problem space*, which is a specialized context within the application domain, in which it makes sense to state a problem.

In other words, it is our opinion that a problem cannot be even stated properly or sensibly without an agreed upon (more often, merely assumed) application domain and problem context. Very often, it is mistakes in the common understanding of this problem context that leads to unexpectedly bizarre or constricting behaviors on the part of the computing system.

So now we have a well-specified problem defined in a problem context. We are purposely setting aside creativity for now, though we believe that this framework can also be applied in that case, with a problem statement of finding the appropriate well-defined problem (this approach is part of our *Problem Pos-*

ing paradigm [20]). Explicitly identifying the problem, and separating it from the possible solutions or required user actions, is an important aspect of our approach. It allows many different possible solution methods to be considered. Since NO one analysis or problem-solving method can deal with all problems in a complex domain [6] [1], it is important to have many methods available.

These form the *resource space*, which contains the computational and information resources that are available to address the problem. It is usually implemented as a large set of independent methods, but we think that more structure here can help (which is why we call it a space).

A certain configuration of those resources is needed to address the particular problem that the user has specified. This collection is usually much smaller than the total resource space, so we call it the *solution space*. Since it contains only those resources required to solve the problem, we would ideally like to have the computing system find this space quickly.

However, in order to find a solution space, very often a much larger *examination space* or *discovery space* must be searched.

For example, in trying to prove a theorem (in geometry, say), the problem space is one in which the assertion can be made, the solution space is one in which the proof can be made, and which often involves extra elements constructed just for the proof. The resource space is the collection of lemmas, theorems, inference rules, problem-solving methods, and previously solved problems, and the solution search space is much wider, since it has to include many different kinds of construction and proof discovery methods.

3. Characteristics

In this section, we discuss the following measurable characteristics of intelligent systems (it can be seen that there are non-trivial overlaps among them, which we try to unravel later on):

1. adaptability,
2. learning,
3. predictive modeling,
4. problem identification,
5. information association,
6. assumptions, and
7. symbolic language.

In each case, we offer an approach to at least one way to compute a measurement value for the characteristic, which we hope will stimulate others to invent and provide better ones.

We make no claim that these are all the important characteristics; discovering others is the point of our research program.

3.1. Adaptability

By far the most commonly expressed attribute of intelligence is *adaptability*, which for us means the speed and scope of adaptability to unforeseen situations, including recognition (of the unforeseen situation), assessment, proposals (for reacting to it), selection (of an activity), and execution. Accurate prediction of effects is even better (and more successful), but we save that one for a later section.

A common example of adaptability is flexible planning, in which a system can react quickly to situations by changing its plans. It seems clear that flexibility in plans is partly the result of their incompleteness: if the detailed goals remain partly unspecified, then there are more possible steps to take. This phenomenon shows up in programming as “late binding”, in which a resource used to address a problem is often not selected until just before it is used (as in our Wrapping approach to heterogeneous system integration in Constructed Complex Systems [19]). The delaying of these decisions does, of course, conflict with rapid execution, and the resulting tradeoff is important and depends essentially on rapid elaboration and evaluations of the choices.

To measure adaptability of a system, we have to present it with different kinds of variability in its environment, and measure its performance, then average that performance over some variability measurement of the environment. The variability in the environment can be static (many different kinds of slowly changing environment), dynamic, (rapidly changing phenomena within the environment), and in both cases, we can describe the degradation in performance as a function of the variability in the environment.

3.2. Learning

Another common attribute of intelligence is *learning*, which for us is the rate of effective learning of observations, behavior patterns, facts, tools, methods, etc. [27]. There is an enormous literature on learning in humans and animals, but our interest here is mainly on the measurements for computing systems that can learn. Learning is about improving performance, so in a sense all of our proposed measurements can be improved by learning. Part of this learning includes concept formation and formulation, which is a way to summarize different structures and processes compactly. We return to this point later on, in the section on symbol systems.

It is important to note here that there are some fundamental limitations on the kinds of symbol systems that can be used

in the expressive tasks above. One of the limitations of any discrete symbol system is the “get stuck” theorems [18] [23], which show that unless a system can change its own basic symbols, and re-express its knowledge and behavior in new symbols, new knowledge gradually becomes harder and harder to incorporate, leading to a kind of stagnation.

Measuring learning is a little easier than measuring adaptability. We have long made a distinction between a *smart* system, which has a lot of knowledge about its domain of applicability, and an *intelligent* system, which can learn new knowledge quickly about its domain of applicability. Smartness is a performance characteristic that is relatively easy to measure, and the ability to learn, which is about improving that performance, is easy but time-consuming to measure.

3.3. Predictive Modeling

An important way to be less surprised at environmental phenomena is *predictive modeling*, which for us means accurate modeling and prediction of the relevant external environment. This kind of modeling includes the ability to make more effective abstractions (which is treated below in a later section). Since a system cannot know everything about its environment, we assume that there will be multiple models carried in parallel, with new data interpreted into information using the model as an interpretive context, and each model adjusted, assessed, and ranked for likelihood continually. This kind of modeling makes the computing system an *anticipatory system* in the sense of Rosen [33], since it can make current decisions on the basis of its models of future effects of its decisions. It is therefore expected to be much more capable than a merely reactive system, since it can be preparing responses to its environment before anything important happens in the environment.

A concrete example of this kind of modeling is trying to distinguish trends from fluctuations at different time scales in a complex environment. In such an environment, activity occurs at many time scales, so the only viable approach is multiresolutional [31] [32], that is, the system must maintain several different filtering processes that examine the environment at different resolutions (time, space, and even conceptual), and look for local stationarity.

There are three kinds of models to be considered: empirical models, which are computed according to the observed data, *a priori* models, which are provided up front, and fitted to the data (we think these are much less important than the others), and deduced models, which are derived from other models and knowledge available.

In addition, analyses of these models requires several different kinds of reasoning, both mathematical and linguistic [16]. These methods include *case-based reasoning*, in which the system tries to match the current situation with one it has encoun-

tered before, *deductive reasoning*, which can be illustrated as having statements “A” and “A implies B” and concluding statement “B”, and *inductive reasoning*, which can be illustrated as having statements “A” and “B” and concluding statement “A implies B”. The best-known example of inductive reasoning is *exploratory pattern analysis*, which is a way of extracting properties of mostly unknown data. The last style of reasoning is *abductive reasoning*, which can be illustrated as having statements “B” and “A implies B” and concluding statement “A”. This style of reasoning is the one corresponding to explanation, since it follows the deductive chains backwards.

Measuring the modeling capability is not about comparing the resulting models with the processes underlying the environmental phenomena, but rather, it is about measuring the correctness or appropriateness of the predictions. Some predictions take the form “this phenomenon is unimportant”, while some must be much more definite, such as “the moving ball will be there at that time” or “the closing door will be open enough for a few seconds”. Once explicitly formulated, these predictions can be compared, and the results plotted against the complexity of the prediction task (which we as evaluators must assess).

3.4. Problem Identification

The best way to respond to problems quickly is to identify them quickly, which requires speed and clarity of problem identification and formulation. In our opinion, speed of problem solution is secondary. Even if we seem to specify a problem as a constrained search, we seem to construct search spaces that are very problem-specific, often extremely intricate, constructed using the constraints directly (i.e., not by searching a large encompassing space, and ignoring the parts outside the constraints).

This problem identification problem is a special case of the situation identification problem, in which acceptable performance is often dependent on recognizing that a situation is similar to one encountered before, and that, in turn, depends on identifying the “right” set of features of the situation to explicitly notice and recall.

Naive models of situated computing systems assume that all of the important data that defines a situation is contained in the sensor values for that instant. Humans don’t do that; we seem to extract information from the data, based on a number of continual, particular, and only occasionally goal-directed models, and retain only a small part of the actual sensor data. There is also some reason to believe that we only keep interval averages, not instantaneous pictures, of a situation (even a mental image is the result of a lot of processing, for object separation and identification, etc.).

The ability to identify important situation features quickly and correctly depends on having at hand the right specification spaces to determine and describe the features.

Very often, the application domain and problem context that allow a problem or even a situation to make sense must be inferred from the observable environmental behavior. This process is also part of good problem identification, a kind of recognition or noticing.

Good problem identification is an intermediate stage between goals and solutions, so it must in part depend on the resources available to a system.

Criteria for good problem identification are still difficult to describe. We will take speed of problem formulation, succinctness of problem statement, and accuracy of problem statement to be the main criteria. Here, we can only assess the accuracy of the problem statement using knowledge of the potential solution methods, since the effectiveness of the problem statement depends on which resources can address it.

3.5. Association

One of the clearest signs of intelligence is the wide scope and effectiveness of associations, and the corresponding evaluation of disparate information for inclusion into a decision process. Discovery and explanation of new associations is even frequently associated with creativity.

This includes several different kinds of reasoning, from analogies and use of metaphors, through the connection of facts to inference rules. It includes ways to use complex relationships summarized numerically (as we so often do when we implement these systems), and it must include a very flexible reasoning system [16]. There is some argument to the effect that all of these can be viewed as similarities in conceptual spaces [10], as long as we make the class of spaces large enough (i.e., not just numerical ones).

These abstract associations are also part of the mysterious phenomenon of “noticing”, which can occur when repeated or anomalous environmental effects are pushed into awareness, seemingly without any prior attention. Similarly, we seem to be adept at noticing correlations in temporal sequences (this ability clearly has some evolutionary advantages), even when they occur in distinct sensory or conceptual spaces.

The simplest version of these processes uses empirical statistical techniques, such as the use of co-occurrence measurements in natural language information retrieval. These and related methods work surprisingly well for this case [26], and we have shown that they can be used in other areas as well [12] [24].

On the other hand, what allows these methods to work well is the explicit representations for words and phrases in the kinds of documents used. In our case of Constructed Complex Systems, the system has to make the representations explicit first, after which the analyses are relatively easy. In particular, it

is important to have a representational mechanism that allows comparisons in many different conceptual spaces, so that different kinds of associations can be computed and analyzed.

Since we discuss in other sections the choices of representation and the difficulties of appropriate ones, we consider in this section only the problem of computing associations. We could posit that the wider the associations range, i.e., the more conceptual spaces are involved, the better the association process, but that width of scope has to be traded off against the speed of use of the associations, since we are actually only able to measure performance, not competence. This ability will manifest itself as an improved ability to recognize similarities in difficult problems, and an improved ability to use unlikely resources to address problems in a useful way.

3.6. Assumptions

A perennial problem with reasoning in systems, and particularly with deduction, is the mis-identification and conflation of assumptions. It is important that a system can identify its more important assumptions and prerequisites, which includes the ability to widen a context (by removing some of the assumptions).

This problem is a special case of Computational Reflection [28] [11], which is the ability of a Constructed Complex System to analyze its own behavior [15]. Having access to internal data structures and reasoning processes in an explicit and analyzable way allows a system to monitor its own behavior, short-circuit unsuitable lines of reasoning, and perform “what-if” studies of itself, which can eliminate some errors before they occur [21] [24]. We have shown that it is relatively straightforward to implement systems with this kind of Computational Reflection [17], but the general case is much harder.

We can consider systems that identify the prerequisites of an action, since identification of prerequisites is abductive reasoning (also called “backward chaining” in the Artificial Intelligence literature), but designing a system that can determine a context limitation, which is a kind of prerequisite of representation, and then move outside that limitation, is much harder.

Identifying assumptions is a kind of creative reasoning, that examines reasoned arguments and transform them into an identification of the assumptions and inference rules required to accomplish the arguments. Since we expect the system to perform these operations itself, it must have a mechanism for reasoning within a system, about the boundaries and limitations of that system. We think that this ability is both hard and essential for intelligent systems.

We can measure how well a system identifies its own assumptions by placing it into environments where many common assumptions fail, and checking how well the system per-

forms. We can also use environments in which the basic assumptions change with time, to see if the system can react sufficiently quickly. These measurements are subtle, and disentangling them from the other possible reasons for performance failures will be difficult. We need much better measurements here.

3.7. Symbolic Language

Perhaps the most important property of all, in our opinion, is the use of symbolic language for explicit representations, including the range and use of analogies and metaphors (this is about identification of similarities), and the invention of symbolic language, which includes creating effective notations for internal representation. This property is not altogether unchallenged, but despite the “behavior-based” intelligence work [29], we believe representation to be essential at all levels of intelligence [3], especially for computing systems.

We repeat here that we don’t care particularly whether living systems (and in particular humans) have all of these models explicitly represented or implicitly embodied. Our Constructed Complex Systems will have them all explicit.

This property should be unraveled into several different characteristics, but there doesn’t seem to be an appropriate analysis of it yet, though there are some promising or at least interesting approaches [34] [7], and we have proposed an architecture that emphasizes the symbol systems [22].

Such an approach to the use of symbols in Constructed Complex Systems must account for the semantics of representation [35], at many different levels, and for the processes that change those representation methods (our conceptual categories are an example representation style [13] [14], and our computational semiotics research is about changing the symbol system when it becomes necessary [18] [23]).

It turns out that human expertise often correlates with better-organized knowledge, and not just with more knowledge, so that problems are recognized more quickly [8].

Since, in our opinion, appropriate abstraction requires a repertoire of conceptual spaces, so that the important properties of the situation at hand can be matched to many more choices of analysis space, and evaluative assessments can become part of the matching process, we think that a very large repertoire is needed, together with some very flexible and fast indexing methods.

Following our own symbol system studies here [13] [14], we measure the use of symbol systems via an efficiency notion: the total size of the representations used compared to the scope of what is represented. This comparison can be estimated using the analysis described in the papers cited: a fixed symbol system has a fixed finite set of basic symbols, and a fixed finite set of

symbol structure combination methods. These sets strictly limit the number of distinctions that can be represented within the symbol system with each size expression. If the system can also change the combination methods, then the numbers can be much larger (though they are still computable).

This measurement is, of course, an intrinsic one (i.e., it is a competence measure), not an extrinsic one (i.e., a performance measure), but we think that it will help us develop more performance measurements. In addition, we want some other performance measurements, such as the speed of representational encoding, measured in some units independent of machine-hardware, and the speed of interpretation of those representations (which is about determining the appropriate action to take). There are many other possible measures here.

4. Intelligent Systems

In this section, we discuss how these issues affect the design of Constructed Complex Systems [15], which are artificially constructed systems that are managed or mediated by computing systems. We are concerned with issues of autonomous and intelligent behavior in such systems, which for us, at least means that the system takes a major role in selecting its own goals [17] [25]. When we expect Constructed Complex Systems to operate autonomously, whether out in the real world or in cyberspace, we need to incorporate a great deal of flexibility and adaptability into their design and implementation. We have shown one way to implement such a system [21] [22], one that also helps avoid the most common difficulties found in complex computing systems: rigidity and brittleness.

Biological systems have much more flexible and powerful adaptation properties than most constructed systems [5], and a careful consideration of their properties provides stringent requirements for the kind of Constructed Complex Systems that would be able to act autonomously. It also gives us some hints about the design structures that are needed [30] [17].

Our approach is to define a new kind of architecture [22] that includes both our Wrapping integration infrastructure [19] and our Problem Posing interpretation [20], that provides a declarative interpretation of all programming languages, so that posed problems can be separated from applicable resources, and our conceptual categories [13] [14] to provide a flexible representation mechanism that separates model structures from the roles they play.

Our Wrapping architecture provides the required flexibility by supporting systems that are variable as far down as we choose to make them (even all the way down through the operating system to the hardware) [15]. One reason that we want this variability is that we expect to study many different approaches to any given problem area, and our infrastructure has to support alternatives for almost every part of every process. In fact, one

of the principles we have highlighted in our architecture investigations is that NO one model, language, or method suffices for a complex system (or environment), so the variability is not just convenient; it is necessary [6] [1].

In addition, we take the hypothesized common origin of language and movement [3] as a hint, since the implied layers of symbol systems can be implemented easily in Constructed Complex Systems using a meta-level architecture [17].

In addition to the data and processes, we also need a third style of computation, that of “re-expression”, which allows a system to re-organize itself when its current organization is not adequate. What this means for us is that the system can somehow detect when its own representational mechanisms are not adequate, and it can use the failures to help invent new ones.

To make things even more interesting, we also want to have the system decide for itself when it needs to be re-organized, because its fundamental symbol systems are not expressive or powerful enough, and then carry out for itself the re-organization automatically, by defining new symbol systems and re-expressing itself in the new terms. This behavior is hard to implement usefully, but we have made some progress in identifying the important issues.

The Wrapping processes give the process structure and the Wrappings and conceptual categories give the data structure. The re-expression criteria are implemented as resources that monitor the system. We describe each of these technical issues in turn, and then show how they can be used to help construct the kind of system we want to build.

The essence of computation is interpretation of symbol systems. The only operations that a digital computer can perform are copying and comparison. All arithmetic in digital computers is via limited-precision explicit models of the corresponding integer or real arithmetic. Therefore, we cannot construct computing systems to do complex or otherwise interesting tasks without many explicit models of the kinds of computation, deduction, or analysis required. All of these models must then be expressed in terms of the operations that we can implement on these (very) limited computers.

The theorems of Turing, Gödel, and others show that there are fundamental limits on the expressive and computational power of computing systems, but ALL of the theorems assume that the symbol system remains fixed (that is a basic assumption in all of the mathematical proofs), and that the parallelism can be mapped into interleaved events. Systems that are not restricted in either of these ways might escape the bounds of these theorems. This is one of our current direction of research [18] [22] [23].

5. Conclusions

We care about measuring intelligence because we want to build such devices, and without some better measurement processes, we will have no repeatable way to evaluate and compare different designs.

We have described some properties that we think are important, that have driven our research in Constructed Complex Systems, including a few that have not been extensively used or identified in the literature. We do not think that they completely cover the spectrum of what is commonly considered to be intelligent behavior, but they do cover more of the scope than simply “adaptability” or “intellect”.

We have examined these properties to determine what they require as fundamental enabling capabilities, and described an architecture that includes all of these enablers, as a way to test our assertions about the connection between them and intelligent behavior. We expect that as we build systems with more of these enablers, the systems will exhibit more of the important properties we have identified, and at the same time they will seem more intelligent.

We think that this problem is hard, and that we are on a right track (we make no assumption about how many right tracks there may be; the more we collectively explore, the more likely it is that we will get some of the right answers). We think that fundamental investigations like these are necessary; we hope that they are sufficient.

References

- [1] Kirstie L. Bellman, “An Approach to Integrating and Creating Flexible Software Environments Supporting the Design of Complex Systems”, pp. 1101-1105 in *Proceedings of WSC '91: The 1991 Winter Simulation Conference*, 8-11 December 1991, Phoenix, Arizona (1991); revised version in Kirstie L. Bellman, Christopher Landauer, “Flexible Software Environments Supporting the Design of Complex Systems”, *Proceedings of the Artificial Intelligence in Logistics Meeting*, 8-10 March 1993, Williamsburg, Virginia, American Defense Preparedness Association (1993)
- [2] Kirstie L. Bellman, “Sharing Work, Experience, Interpretation, and maybe even Meanings Between Natural and Artificial Agents” (invited paper), pp. 4127-4132 (Vol. 5) in *Proceedings of SMC '97: the 1997 IEEE International Conference on Systems, Man, and Cybernetics*, 12-15 October 1997, Orlando, Florida (1997)
- [3] Kirstie L. Bellman and Lou Goldberg, “Common Origin of Linguistic and Movement Abilities”, *American Journal of Physiology*, Volume 246, pp. R915-R921 (1984)
- [4] Kirstie L. Bellman, Christopher Landauer, “A Note on Improving the Capabilities of Software Agents” (poster summary of [17]), pp. 512-513 in *Proceedings of AA'97: The First International Conference on Autonomous Agents*, 5-8 February 1997, Marina Del Rey, California (1997)
- [5] Kirstie L. Bellman and Donald O. Walter, “Biological Processing”, *American Journal of Physiology*, Volume 246, pp. R860-R867 (1984)
- [6] Richard Bellman, P. Brock, “On the concepts of a problem and problem-solving”, *American Mathematical Monthly*, Volume 67, pp. 119-134 (1960)
- [7] Terrence W. Deacon, *Symbolic Species: The Co-Evolution of Language and the Brain*, Norton (1997)
- [8] K. Anders Ericsson, Reid Hastie, “Contemporary Approaches to the Study of Thinking and Problem Solving”, Chapter 2, pp. 37-79 in [36]
- [9] Kenneth D. Forbus, Johann de Kleer, *Building Problem Solvers*, A Bradford Book, MIT Press (1993)
- [10] Peter Gärdenfors, *Conceptual Spaces: The Geometry of Thought*, MIT (2000)
- [11] Gregor Kiczales, Jim des Rivieres, Daniel G. Bobrow, *The Art of the Meta-Object Protocol*, MIT Press (1991)
- [12] Christopher Landauer, “Correctness Principles for Rule-Based Expert Systems”, pp. 291-316 in Chris Culbert (ed.), *Special Issue: Verification and Validation of Knowledge Based Systems, Expert Systems With Applications Journal*, Volume 1, Number 3 (1990)
- [13] Christopher Landauer, “Conceptual Categories as Knowledge Structures”, pp. 44-49 in A. M. Meystel (ed.), *Proceedings of ISAS'97: The 1997 International Conference on Intelligent Systems and Semiotics: A Learning Perspective*, 22-25 September 1997, NIST, Gaithersburg, Maryland (1997)
- [14] Christopher Landauer, “Conceptual Categories in the Information Infrastructure”, in *Proceedings of ICC'98: 1998 International Congress on Cybernetics*, 24-27 August 1998, Namur, Belgium (1998)
- [15] Christopher Landauer, Kirstie L. Bellman, “Constructed Complex Systems: Issues, Architectures and Wrappings”, pp. 233-238 in *Proceedings of EMCSR'96: Thirteenth European Meeting on Cybernetics and Systems Research, Symposium on Complex Systems Analysis and Design*, 9-12 April 1996, Vienna, Austria (April 1996)
- [16] Christopher Landauer, Kirstie L. Bellman, “Mathematics and Linguistics”, pp. 153-158 in Alex Meystel, Jim Albus, R. Quintero (eds.), *Intelligent Systems: A Semiotic Perspective, Proceedings of the 1996 International Multidisciplinary Conference, Volume I: Theoretical Semiotics, Workshop on New Mathematical Foundations for*

Computer Science, 20-23 October 1996, NIST, Gaithersburg, Maryland (1996)

- [17] Christopher Landauer, Kirstie L. Bellman, "Computational Embodiment: Constructing Autonomous Software Systems", pp. 42-54 in Judith A. Lombardi (ed.), *Continuing the Conversation: Dialogues in Cybernetics, Volume I, Proceedings of the 1997 ASC Conference*, American Society for Cybernetics, 8-12 March 1997, U. Illinois (1997); poster summary in [4]; *Cybernetics and Systems: An International Journal*, Volume 30, Number 2, pp. 131-168 (1999)
- [18] Christopher Landauer, Kirstie L. Bellman, "Situation Assessment via Computational Semiotics", pp. 712-717 in *Proceedings of ISAS'98: the 1998 International MultiDisciplinary Conference on Intelligent Systems and Semiotics*, 14-17 September 1998, NIST, Gaithersburg, Maryland (1998)
- [19] Christopher Landauer, Kirstie L. Bellman, "Generic Programming, Partial Evaluation, and a New Programming Paradigm", Paper etspi02 in *32nd Hawaii Conference on System Sciences, Track III: Emerging Technologies, Software Process Improvement Mini-Track*, 5-8 January 1999, Maui, Hawaii (1999); revised and extended version in Christopher Landauer, Kirstie L. Bellman, "Generic Programming, Partial Evaluation, and a New Programming Paradigm", Chapter 8, pp. 108-154 in Gene McGuire (ed.), *Software Process Improvement*, Idea Group Publishing (1999)
- [20] Christopher Landauer, Kirstie L. Bellman, "Problem Posing Interpretation of Programming Languages", Paper etecc07 in *Proceedings of HICSS'99: The 32nd Hawaii Conference on System Sciences, Track III: Emerging Technologies, Engineering Complex Computing Systems Mini-Track*, 5-8 January 1999, Maui, Hawaii (1999)
- [21] Christopher Landauer, Kirstie L. Bellman, "Computational Embodiment: Agents as Constructed Complex Systems", Chapter 11, pp. 301-322 in Kerstin Dautenhahn (ed.), *Human Cognition and Social Agent Technology*, Benjamins (2000)
- [22] Christopher Landauer, Kirstie L. Bellman, "Architectures for Embodied Intelligence", pp. 215-220 in *Proceedings of ANNIE'99: 1999 Artificial Neural Nets and Industrial Engineering, Special Track on Bizarre Systems*, 7-10 November 1999, St. Louis, Mo. (1999)
- [23] Christopher Landauer, Kirstie L. Bellman, "Symbol Systems in Constructed Complex Systems", pp. 191-197 in *Proceedings of ISIC/ISAS'99: International Symposium on Intelligent Control*, 15-17 September 1999, Cambridge, Massachusetts (1999)
- [24] Christopher Landauer, Kirstie L. Bellman, "Detecting Anomalies in Constructed Complex Systems", in *Proceedings HICSS'2000: The 33rd Hawaii International Conference on System Sciences, Track IV: Emerging Technologies*, 4-7 January 2000, Maui, Hawaii (2000)
- [25] Christopher Landauer, Kirstie L. Bellman, "Reflective Infrastructure for Autonomous Systems", in *Proceedings of EMCSR'2000: The 15th European Meeting on Cybernetics and Systems Research, Symposium on Autonomy Control: Lessons from the Emotional*, 25-28 April 2000, Vienna (April 2000)
- [26] Christopher Landauer, Clinton Mah, "Message Extraction Through Estimation of Relevance", Chapter 8, in R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, P. Williams (eds.) *Information Retrieval Research, Proceedings of the Joint ACM and BCS Symposium on Research and Development in Information Retrieval*, June, 1980, Cambridge University, Butterworths, London (1981)
- [27] Pat Langley, *Elements of Machine Learning*, Morgan-Kaufmann (1996)
- [28] Pattie Maes, D. Nardi (eds.), *Meta-Level Architectures and Reflection, Proceedings of the Workshop on Meta-Level Architectures and Reflection*, 27-30 October 1986, Alghero, Italy, North-Holland (1988)
- [29] Maja J. Mataric, "Behavior-Based Control: Main Properties and Implications", pp. 46-54 in *Proceedings IEEE International Conference on Robotics and Automation, Workshop on Architectures for Intelligent Control Systems*, May 1992, Nice, France (1992)
- [30] Maja J. Mataric, "Studying the Role of Embodiment in Cognition", pp. 457-470 in *Cybernetics and Systems*, special issue on *Epistemological Aspects of Embodied Artificial Intelligence*, Volume 28, Number 6 (July 1997)
- [31] Alex Meystel, "Multiresolutional Architectures for Autonomous Systems with Incomplete and Inadequate Knowledge Representations", Chapter 7, pp.159-223 in S. G. Tzafestas, H. B. Verbruggen (eds.), *Artificial Intelligence in Industrial Decision Making, Control and Automation*, Kluwer (1995)
- [32] Alex Meystel, *Semiotic Modeling and Situation Analysis: An Introduction*, AdRem, Inc. (1995)
- [33] Robert Rosen, "Anticipatory Systems in Retrospect and Prospect", *General Systems Yearbook*, Volume 24, p. 11 (1979); reprinted as Paper 28, pp. 537-557 in George J. Klir, *Facets of System Science*, Plenum (1991)
- [34] Brian H. Ross and Thomas L. Spalding, "Concepts and Categories", Chapter 4, pp. 119-148 in [36]
- [35] John Sowa, *Knowledge Representation*, Brooks / Cole, Pacific Grove, CA (2000)
- [36] Robert J. Sternberg (ed.), *Thinking and Problem Solving*, Academic Press (1994)

Generalizing Natural Language Representations for Measuring the Intelligence of Systems

A. Meystel
Drexel University, Philadelphia, PA 19104

Enhanced abstract

In the core of this method of intelligence evaluation, there is a concept of using natural language as the least damaging medium for representing knowledge of the systems. The goal of all existing methodologies of knowledge representation boils down to performing generalization of this knowledge in one of the existing forms: analytical representation, automata theory, predicate calculus of the first order. Connectionist schemes are not on this list because the problem of generalization upon the entity-relational network (ERN) have not been addressed consistently. In this paper, the concept of constructing a nested multiresolutional system of ERNs by consecutive generalization of them bottom-up and consecutive instantiation of them top-down. It is demonstrated that given a set of problems to be resolved, one can learn which one the nested ERN alternatives is more appropriate for solving this set. Finally, a problem of evaluating ERN “for any set of problems” is discussed.

Conceptual Paradigm. This theoretical paradigm relates to digital text processing equipment such as a computer system used for text processing. The method and the apparatus is used to obtain a structure of text organization, elements of which can be used upon the initial narrative for the subsequent processing in order to generate a variety of different texts that have different degree of compression. It is anticipated that by constructing a proper organization of the text representation, obtained from the original document, different structures of the text could be constructed, for example, the one that would allow to encode its meaning as a set of nested and interrelated generalizations. In turn, this should allow for generating the narrative text from each of these structure. These texts should be different in their level of generalization, focus of attention, and the depth of detail.

General Vision. As soon as the analysis starts, the whole texts changes its initial shape and demonstrates a multiplicity of potential interpretations at each level of resolution. The text subjected to the process of analysis demonstrates its semantic fuzziness, and combinatorial clouds of combinatorial possibilities emerge around each unit of the texts. These "clouds" characterize the interpretational ambiguity which should be eliminated (or at least, substantially reduced) as a result of text processing. The fuzzy and not totally disambiguated units have frequently an emergent property of sticking together, forming new generalized units that precipitate from the fuzzy intermediate structure. Eventually a new text emerges which is shorter than the initial one. This "sticking together" happens in a strictly multi-granular fashion. Each text has a potential to several rounds of compression by generalization as well as to several rounds of enhancement by instantiation. Construction of such a multigranular structure can be performed for each text, or a group of texts.

This process of combinatorial fuzziness generation including the formation of the links of nestedness, and precipitation of the multigranular text structure (together with levels of enhancement and/or compression) can be done spontaneously during free, or goal driven interpretation of an unknown document. However, if the assignment contains the

description of a specific customer's interests, this combinatorial fuzziness generation can be guided by this assignment. It does not necessarily need to be guided. In the latter case, a summary of the general (non-goal-oriented) form is created.

Multiresolutional Structure of Text Representation. Extracting the multiresolutional (multigranular, multiscale) structure (nested hierarchical architecture) of text units (entities) from the Text is a prerequisite to transformation from the narrative representation into the relational architecture of knowledge. The main dictionary is used for the initial interpretation of the units of Text, and the new domain dictionaries are formed for the text-narrative, or Original Text (OT) together with its Structure of Text Representation (STR) as a part of the text analysis. The multiresolutional hierarchy of STR consists of the units, which lump together elements of the text, that has emerged due to the "speech-legacy" grammar. Since the transformation of OT into STR can be done through incremental generalizations within OT, building the vocabulary of the OT is a prerequisite for the subsequent STR construction.

The vocabulary is a list of "speech-legacy" words that are symbols for encoding entities of the real situations and can be represented by single words as well as groups of words. *Entity* is defined as a thing that has definite, individual existence in reality or in the text; something "real" by itself. In other words, an entity of the reality is anything that exists, important for registering and memorizing, has a meaning as a part of some functional description and is (or should be) assigned a separate word (or a group of words) no matter whether we use it as a part of "speech-legacy" representation, or an element of the STR. The first problem to be resolved is finding entities that are represented by single words, then test groups of interrelated words, as phrases that denote entities. Therefore, functioning of STR requires understanding how

Units of representation. The decomposition of the uniform chaotic informational medium takes place driven by the initial goal and a set of criteria that might determine different kinds of uniform media. Thus, the results of developing the linguistic world representation depends on the aspect of interest submitted and encoded by the user. As a result of recognition processes, a variety of singular information units (entities) emerges, which fit within a natural categorization that is implicitly influenced by the observer. Formation of singularities (as entities) can be metaphorically described as a result of clustering processes in which the elementary units of the primordial Text gravitate to each other in the areas of higher informational density (where the elementary units are more in quantity, more interrelated, and more important for the user. For clarifying the gravitational metaphor, we should emphasize that for the further discussion it is irrelevant whether the density is increased as a result of the gravitation, or gravitation starts prevailing because of an initial increase in density. In our disclosure, these processes are to be understood in computational terms. At this point, the observer will legitimately appear in our presentation as a carrier of the interrelated concepts: scale, resolution, and granulation.

The concept of scale allows for introduction of a formidable research tool that can be applied for each couple of adjacent levels of knowledge organization obtained by the method described in this Section of the disclosure. This tool is related to the specifics of a different interpretation of units in higher and lower levels of resolution (HLR and LLR). The units of the HLR emerge as a result of the process of forming singularities at the previous, even higher resolution level (which is not a part of our couple levels of resolution under consideration). After these singularities have been formed, they receive an interpretation, a meaning, a separate word of a vocabulary at this HLR. For the LLR of the pair of levels that we discuss, these particular singularities have no meaning at all.

The meaning will emerge after these entities of the higher level of resolution (HLR) will assemble together into a singularity which can be recognized by the user at the lower resolution level (LLR) as a meaningful entity. Before grouping

of these entities into meaningful singularities happens, they are just nameless units with a tendency to gravitate to each other, expressed in the set of their relations. This phenomenon is similar to physical gravitation although the gravitation “force” depends on the text, context, goals, and other details of the situation. So, the process of entity formation for LLR recognizes the entities of a HLR just as a set of anonymous units. Their “gravitational” field leads to clustering of features and can give a birth to a new entity of LLR.

Phenomena of Attention: Scope and Focus. Windowing that has been demonstrated in Figure 3 is a result of the need to focus our attention within a specific scope. Let us consider a particular zone of the medium that we use to evaluate; we will call it *the scope of interest*. An imaginary large window (*the scope of attention*) is to be imposed upon the medium (*scope of interest*). Then, the smaller window is sliding within the scope of interest to evaluate the information density. Thus, the size of the scope of attention is presumed to be substantially smaller than the scope of interests. Density of non-uniform units is to be computed within this window which allows evaluation of the continuum quantitatively. Then the window slides over the whole scope of interest, and in each position the density is again computed.

The sliding strategy of moving the window of attention over an Image and/or Text is assigned in such a way that all scope of interest can or will be investigated efficiently. This strategy can be different for constructing different models: we can scan it in a parallel manner; we can provide a very unusual law of scanning; we can make random sampling from different zones of the scope of interest. The strategy selection should depend on needs, hardware tools, and resources available (for example, time). If values of density are about the same everywhere (with small variations within some particular interval) then the medium is considered to be uniform.

Notice, a) that in order to introduce the concept of uniformity we used a sliding window which is one of the techniques of *focusing attention*; b) that in order to form entities of a particular level of resolution we should *group* the entities of the higher level of resolution; c) that to find candidate units for grouping we should *search* for future members of these groups or otherwise *combine* them together. Later we will return to these operations as components of the elementary unit of intelligence.

The idea of meaning of representation closure (MRC-loop) is tantamount to two fundamental ideas: a) the cybernetic idea of feedback or circularity of control (information) circulating in the locally closed system to provide the ability for the system to meaningfully function, and b) the physical idea of energy and matter conservation in the closed system.

The phenomenon of closure can be demonstrated required for explication of the meaning of representation which provides for complete informational connectedness for the flow of representation which starts and ends in the Virtual WORLD represented within a level of knowledge architecture. It is held at each level of granularity for the virtual World determined by the alternatives of meaning implied by the Text and the Goal of the user.

We have developed algorithms of bottom-up consecutive generalization of ERN that represent the text and algorithms of top-down instantiation. We have applied these algorithms to realistic text. The advantages as well as deficiencies of the original text demonstrate itself more graphically as the process of generalization develops. As a result, it becomes possible to judge the original text and even evaluate its advantages and disadvantages quantitatively.

Towards Measures of Intelligence Based on Semiotic Control *

Dr. Cliff Joslyn

Computer Research and Applications Group (CIC-3)
MS B265, Los Alamos National Laboratory
Los Alamos, NM 87545
joslyn@lanl.gov, <http://www.c3.lanl.gov/~joslyn>

July, 2000

Abstract

We address the question of how to identify and measure the degree of intelligence in systems. We define the presence of intelligence as equivalent to the presence of a control relation. We contrast the distinct atomic semiotic definitions of models and controls, and discuss hierarchical and anticipatory control. We conclude with a suggestion about moving towards quantitative measures of the degree of such control in systems.

1 Introduction: A Control Theory Framework for Intelligence

We consider some of the challenges presented in the white paper designed to prepare for this conference [13]. I take the fundamental question to be “How can we as external observers measure the degree of intelligence in a target system?”

One approach is to invoke the typical lists which can characterize intelligent behavior, including adaptability, complexity of internal models, problem solving ability, etc. But what is fundamental to each of these? For example, adaptability is the ability to adjust responses to make them appropriate under variable conditions. Problem solving is the ability to come to

a correct choice about actions to achieve a particular goal, hereby solving the problem. And finally, complexity of internal models must always be considered as relative to their ability to predict the outcome of future behaviors.

Thus can see that fundamental to all of these is the idea that intelligence requires the ability of a system to make *appropriate decisions given the current set of circumstances* [1, 2, 3]. On analyzing this a bit further, we can identify the following necessary components:

Measurement: The ability to know the current set of circumstances.

Decision: The freedom to choose between one of many possibilities.

Goal: The possibility that the choice made will be either appropriate or inappropriate relative to a goal state.

Action: The ability for the decision to affect external and future events, in order for them to be either closer to or further away from the goal.

2 Intelligence as Semiotic Control

We note the similarity to the scheme of an intelligent system as outlined in the conference White

*Prepared for the 2000 Workshop on Performance Metrics for Intelligent Systems.

Paper [13]. This requires a “loop of closure” consisting of six modules: a world interface, sensors, perception, a world model, behavior generation, and actuation. We understand this situation as the existence of a *semiotic control system*. We know briefly outline the theory of semiotic systems.

2.1 Semiotic Models and Controls

There is a rich literature (eg. [5, 15, 17, 18, 19]), traceable back to the founders of systems theory and cybernetics in the post-war period [4], which has tried to construct a coherent philosophy of science based on two fundamental concepts:

- **Models** as the basis not only for a consistent epistemology of systems, but also as an explanation of the special properties of living and cognitive systems.
- **Control systems** as the canonical form of organization involving purpose or function.

While controls and models are distinct kinds of organization, what they share is a common basis in semiotic processes, in particular the use of a measurement function to relate states of the world to internal representations. Perhaps for this reason there has been some ambiguity in the literature about the specific nature of controls and models, and more importantly how they interact. This has led to confusion, for example, about the role of feedback vs. feedforward control, and endo-models *within* systems vs. exo-models *of* systems.

Consider first a classical control system as shown in Fig. 1. In the world (the system’s environment) the dynamical processes of “reality” proceed outside the knowledge of the system. Rather, all knowledge of the environment by the system is mediated through the measurement (perception) process, which provides a (partial) representation of the environment to the system. Based on this representation, the system then chooses a particular action to take in the world, which has consequences for the change in state

of the world and thereby states measured in the future.

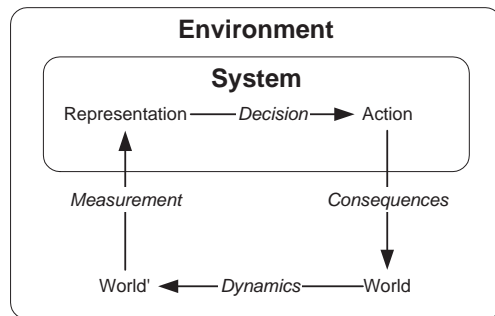


Figure 1: Functional view of a control system.

To be in good control, the overall system must form a negative feedback loop, so that disturbances and other external forces from “reality” (for example noise or the actions of other external control systems) are counteracted by compensating actions so as to make the measured state (the representation) as close as possible to some desired state, or at least stable within some region of its state space. If rather a positive feedback relation holds, then such fluctuations will be amplified, ultimately bringing some critical internal parameters beyond tolerable limits, or otherwise exhausting some critical system resource, and thus leading to the destruction of the system as a viable entity.

Now consider the canonical modeling relation as shown in Fig. 2. As with the control relation, the processes of the world are still represented to the system only in virtue of measurement processes. But now the decision relation is replaced by a prediction relation, whose responsibility is to produce a new representation which is hypothesized to be equivalent (in some sense) to some future observed state of the world. To be a good model, the overall diagram must commute, so that this equivalence is maintained.

As outlined here, models and controls are distinct and atomic kinds of organization. We have argued [8] that this capability begins with living systems, and perhaps defined the necessary and sufficient conditions for living systems.

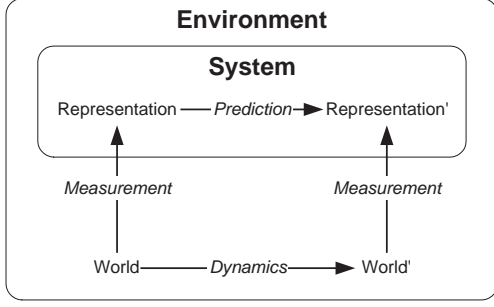


Figure 2: Functional view of the modeling relation.

2.2 Hierarchical Control

Of course, all of the relations described here are a great deal more complex in real intelligent systems. In particular, usually controls and models are considered together. This concept is fully developed elsewhere [7, 9]. We now summarize the primary results of these considerations.

First, the classical view of linear control systems theory [14] is recovered by introduced a “computational” step which plays the role of cognition, information processing, or knowledge development. Typically, extra or external knowledge about the state of the world or the desired state of affairs is brought to bear, and provided to the agent in some processed form, for example as an error condition or distance from optimal state. So now measured states are manipulated and compared to a goal state.

In particular, we are impressed by Bill Powers system for hierarchical control [15, 16, 6], which he has successfully generalized to explain the architecture of neural organisms. As shown in Fig. 3, he views the computer as a comparator between the measured state and a hypothetical set point or reference level (goal). This then sends the second representation of an error signal to the agent. He also explicitly includes reference to the noise or disturbances always present in the environment, against which the control system is acting to maintain good control. For us, these are bundled into the dynamics of the world.

Another great virtue of Powers’ control theory

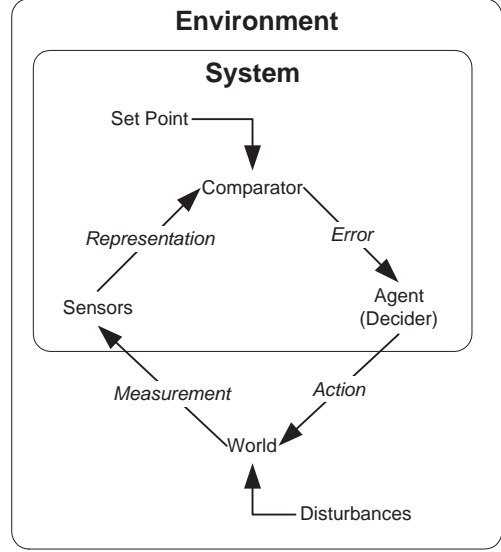


Figure 3: A Powers’ control system.

model is its hierarchical scalability. Fig. 4 shows such a hierarchical control system, containing an inner level 1 and the outer level 2. The first key move here is to allow representations to be combined to form higher level representations. In the figure S_1 and S_2 are low distinct level sensors providing low level representations R_1 and R_2 to the inner and outer levels respectively. But R_1 is also sent to the higher level S_3 , and together they form a new high level representation R_3 .

The second step is the ability for the action of one control system to be the determination of the set-point of another, thus allowing goals to decomposed as a hierarchy of sub-goals. In the figure, the outer level uses R_3 to generate the action of fixing the set point of the lower level. Note how this recovers Meystel *et al*’s “Feature 10” of multiscale knowledge representation where the action of a lower level system is actually the goal of an upper level system [13].

Notice also that the overall topology of the control loop is maintained. While ultimately the lower level is responsible for taking action in the world, it is doing so under the control of the comparison of a high-level goals against a high-level representation. Neural organisms especially are

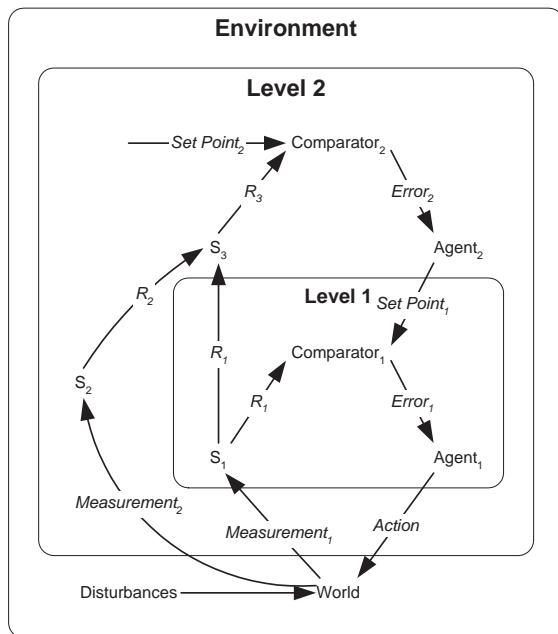


Figure 4: Hierarchical nesting of Powers' control systems.

systems of this type, low-level motor and perceptual systems combining to accomplish very high-level tasks. And of course, determination of the outermost goal is not included within Powers' formal model.

2.3 Anticipatory Control

While familiar to us as a standard engineering discipline, a number of researchers are pursuing the applicability of this kinds of semiotic control [12]. It is also being generalized to a number of other engineering [2] and scientific domains.

However, our normal sense of control combines it with models, which are used to aid in decision-making by predicting future states of anticipated actions, using prediction of future events to guide actions. This is what Ashby refers to as “cause control” [4], or Rosen as “anticipatory” [17], or Klir as feedforward [10]. In this architecture an endo-model embedded *within* a control system is used to make a decision as to which action to take, and thus acts in the role of the agent. It is

this view which most dominates our conception of the nature of control in general.

However, this architecture is actually highly complex and special. It is shown in Fig. 5, where now the agent is replaced by an inner system which is *both* a model and a control system (the arrows have been reflected diagonally to make the graph planar and ease the drawing). This inner system is a control system in the sense that there are states of its “world”, its “dynamics”, and an “agent” making decisions.

However, it is also a model in that the states of its “world” are in fact representations, and its “dynamics” is actually a prediction function. The inner system is totally contained within the outer system, and runs at a much faster time scale in a kind of modeling “imagination”. The representation R from the sensors is used to instantiate this model, which takes imaginary actions resulting in imaginary stability within the model. Once this stability is achieved, then that action is exported to the real world.

Note that the outer control loop here is simple, lacking computation. In Powers' terms, there is no set point which the state of the internal model is being compared to. But this could be present in a slight elaboration where an imaginary measurement is taken from “world” and compared to some set point. The outer error signal would then be fed to change the imagined actions inside the model until stability is achieved.

3 Tests for the Presence of Control

Thus we have now transformed the original question of “how do we measure intelligence?” to “How can we as external observers determine whether a target system manifests control relations with its environment?” and “How can we then measure the degree and modalities of that relation?” I would then offer some ideas based on the work of Powers and his colleague Rick Marken [11, 15, 16].

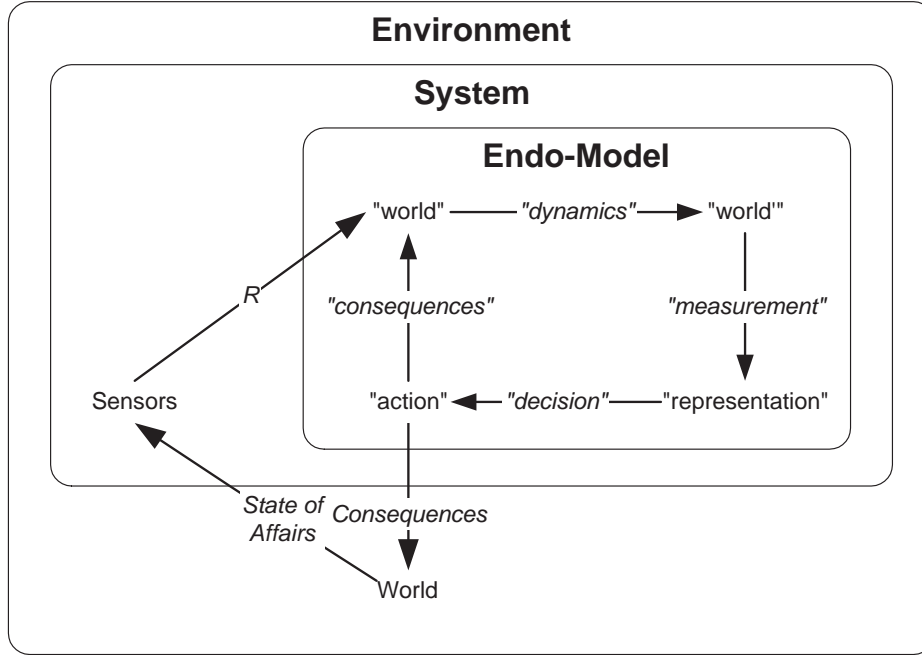


Figure 5: Anticipatory control.

They address the question from the following perspective. Control relations, in virtue of the stability of the controlled variables in the environment, have many of the characteristics of other equilibrium phenomena. Both the thermostat and the ball rolling to a stop at the bottom of a hill evidence this kind of stability behavior. In the first case, the ball does not *want* to roll down the hill, but in a very real sense, the thermostat *does* want to regulate its “perception” of the state of the room temperature.

So how can we distinguish a complex dynamic equilibrium from a control relation? Powers and Marken do this distinguishing on the basis of what they call The Test. It involves the system acting in a way which is counter to physical law: if the ball *failed* to roll down the hill, we’d be surprised, thus we hypothesize that such a ball is manifesting a control relation. Similarly, we would normally expect a room to come to equilibrium with its environment. When it does not, and we believe our dynamical model, then we would hypothesize the presence of a control

device, and we might investigate and discover a thermostat. The “intelligence” of such systems is based on their manifesting a semiotic relation which has been selected by evolution or by designers, allowing the system to “choose” to act counter to physical law.

Now the rub is that this Test thereby requires the prior presence of a model of what the system *should* be doing, so that we can be surprised when it fails to do so. Thus our recognition of a control relation in an exogenous system requires of us an *exogenous* model of reality, whether or not the system has any *endogenous* model itself.

4 Towards a Measure of Control-Based Intelligence

So now, given this semiotic control-based view of intelligence, we wish to go on and attempt to quantify and characterize the degree and kind of control relations present. Thus the problem of measuring intelligence revolves around our ability to measure:

- The amount of phenomena under control;
- The number of environmental distinctions measured by the system;
- The complexity of modalities of measurement and control;
- The complexity of the environmental variety available to the measurement and control of the system;
- If hierarchical control is present, what is the depth of the hierarchy of control; and
- If anticipatory control is present, what is the complexity of the internal, endogenous models?

No doubt in both real and designed systems these are all related to each other in complex ways. However, each of these quantitative terms is effectively a statistical information measure, a measure of variety or freedom. Thus they are amenable to information-theoretical measures like entropies, based on quantities of variety, distinctions, and constraints which a control system can recognize in its environment and then act on in appropriate ways.

References

- [1] Albus, James S: (1991) "Outline of a Theory of Intelligence", *IEEE Transactions on Systems, Man, and Cybernetics*, v. **21**:3, pp. 473-509.
- [2] Albus, James S: (1999) "The Engineering of Mind", *Information Sciences*, v. **117**, pp. 1-18
- [3] Albus, James S: (2000), discussion on email list iab-list@icompsol.com
- [4] Ashby, Ross: (1956) *Introduction to Cybernetics*, Methuen, London, <http://pcp.vub.ac.be/books/IntroCyb.pdf>
- [5] Cariani, Peter A: (1989) *On the Design of Devices with Emergent Semantic Functions*, SUNY-Binghamton, Binghamton NY, PhD Dissertation
- [6] *Control Systems Group*, <http://www.ed.uiuc.edu/csg>
- [7] Joslyn, Cliff: (1995) "Semantic Control Systems", *World Futures*, v. **45**:1-4, pp. 87-123
- [8] Joslyn, Cliff: (1998) "Are Meaning and Life Co-extensive?", in: *Evolutionary Systems*, ed. G. van de Vijver, pp. 413-422, Kluwer
- [9] Joslyn, Cliff: (2000) "The Semiotics of Control and Modeling Relations in Complex Systems", under review for *Biosystems*
- [10] Klir, George: (1991) *Facets of Systems Science*, Plenum, New York
- [11] Marken, Richard S: (1988) "The Nature of Behavior: Control as Fact and Theory", *Behavioral Science*, v. **33**, pp. 196-206
- [12] Meystel, Alex: (1996) "Intelligent Systems: A Semiotic Perspective", *Int. J. Intelligent Control and Systems*, v. **1**, pp. 31-57
- [13] Meystel, Alex, *et al.*: (2000) "Measuring Performance of Systems with Autonomy: Metrics for Intelligence of Constructed Systems", white paper for *2000 Workshop on Performance Metrics for Intelligent Systems*, http://www.isd.mel.nist.gov/conferences/performance_metrics/white_paper.html
- [14] Nise, Norman S: (1992) *Control Systems Engineering*, Benjamin-Cummings, Redwood City CA
- [15] Powers, WT: (1973) *Behavior, the Control of Perception*, Aldine, Chicago
- [16] Powers, WT, ed.: (1989) *Living Control Systems*, CSG Press
- [17] Rosen, Robert: (1985) *Anticipatory Systems*, Pergamon, Oxford
- [18] Rosen, Robert: (1991) *Life Itself*, Columbia U Press, New York
- [19] Turchin, Valentin: (1977) *Phenomenon of Science*, Columbia U Press, New York

Selected Comments on Defining and Measuring of Machine Intelligence

Paul K. Davis

RAND and RAND Graduate School

June 11, 2000

ABSTRACT

This paper records some thoughts about defining and measuring machine intelligence. It touches on (1) the shortcomings of any scalar metric; (2) the power of having mixes of intelligence types in a population of machines; (3) the special issues related to “common sense;” (4) the need to broaden discussion beyond normally understood intelligence; (5) consistent with that, the need in a population to assure for exploration and “mutation;” (6) some technical issues in modeling reasoning in agents; and (7) a methodology (exploratory analysis) for measuring intelligence that emphasizes a diversity of contexts.

Introduction

One of the many lessons learned from a century of work on human intelligence is that intelligence is multifaceted. It therefore appears wise to define and measure machine intelligence as a multidimensional concept.¹

Before elaborating, let me observe that some may quarrel with this conclusion. After all, in many endeavors it has proved feasible to combine various factors into a single scalar quantity that “reasonably” measures what we are interested in. We see this in the applications of multi-attribute utility theory (MAUT)² and in countless modeling problems where people introduce abstractions that combine various factors in ways that appear adequately sound. Furthermore, IQs, SAT scores, and GRE scores are ubiquitous in assessments ranging from the personal (“Wow, Marcus is really smart: he got a double 800.”) to hard-nosed decisions by admissions committees at universities and managers in industry. We all know that intelligence is a complex issue, but most of us nonetheless use the simple metrics—at least to some extent. Moreover, they appear to be more than mere crutches; i.e., they actually do correlate, at least to some extent, with things we care about (e.g., performance in classes or in the business environment). And shorthands are useful.

This said, the quality and depth of any discussion of machine intelligence and its measurement would likely be greatly restricted by having a reductionist goal such as finding a single “IQ.” Talking in shorthand is an excellent way to “dumb down” conversations and inquiries.

Consider the following as part of an indictment:

- The correlation of IQ and SAT scores with subsequent performance in graduate school and life is only very modest. Indeed, it is so modest that one can only puzzle about why so much

¹ A major expositor of the multi-dimensional aspect of intelligence is Howard Gardner.

² The classic reference is Keeney and Raiffa.

fuss is made over the related tests. The answer appears to be only that the scores are the best readily available predictors, even if they are poor predictors.³

- Studies indicate that the predictive power of the simple metrics is particularly poor in explaining, for example, the effectiveness of top executives.⁴
- There are also “glitches” in the tests. Apparently, Richard Feynman’s IQ was rated as “merely” 122.
- At a more personal level, I believe that we probably all know individuals who would solidly flunk tests of mathematics, even simple mathematics, but who are regarded as brilliant in other ways—whether verbally, or, for example, in the arts.
- I suspect that most of us also know individuals well who on the one hand scored very highly on intelligence tests and yet lack the capability to excel in various higher level activities. Perhaps they lack common sense; perhaps they lack creativity; perhaps they are so obsessed with numbers that they cannot deal with fuzzier aspects of life.
- Sallying forth into a more dangerous area, consider now what might be regarded as the unmitigated *stupidity* of famous “geniuses.” In the military domain, one might think of Napoleon, who marched on Moscow in the winter and lost nearly his entire army. Or, to push the debating point even further, what should we make of certifiable sociopaths who happen also to have IQs? Some might claim “Oh, you’re confusing intelligence with mental health.” Perhaps, but which should we care most about when considering the performance of future machines?

Let me now shift from the IQ business to multi attribute utility theory. We know a great deal about its usefulness and shortcomings. Personally, I urge students of policy analysis to *savor* the multiple attributes of strategies and avoid combining them until and unless it is necessary. The paradigm for displaying results of policy analysis is, for most of my colleagues and me, is a “scorecard” in which one views the ratings of options in each of a number of aggregate categories. We may or may not add up the scores for the purpose of having a single, simple-minded, result (e.g., for making cost-effectiveness comparisons), but if we do it is only after we have adjusted assumptions so as to assure that the aggregated result is “right.” By that I mean that decision-analysis methods are often most useful when used iteratively: we try to be logical and explicit; we try to do things by the numbers; we look at the results; we then observe that they are “wrong” (meaning that we don’t like them).⁵ We go back to the assumptions and either fiddle the input scores or muse a bit until we discover some hidden variables that are bothering us, and affecting us implicitly. We then iterate. And so on. At the end of the process, the algorithms may work and we may have a sense that we understand the problem, but this was due to the disaggregated process of getting to that point. At the trivial level, I like to challenge

³ One discussion of this is in Robert Klitgaard’s review of admission criteria. Yet another is in a Robin Dawes’ book.

⁴ It is perhaps of interest to note that the SAT scores of both Presidential candidates have been bandied about in the press. Neither candidates scores appear outstanding when compared to those of top-half applicants to graduate schools. Given the achievements of the individuals to date, doesn’t this tell us something?

⁵ A familiar example here is how most of us read product evaluations in Consumers Reports or PC magazines.

students with a car-buying problem, the purpose of which is to demonstrate that the usual hard-headed approach does a *terrible* job in representing our real values. Some individuals, for example, really do want a red Mercedes, and it's hard to get that answer when looking at mileage, repair costs, etc. etc. Even if one has a category for "prestige" or some such, it is very difficult to get the red Mercedes as the answer unless one essentially zeros out the other categories or recognizes the shortcomings of the MAUT methodology with its assumptions of linearity and related substitutability. On the other hand, it can usefully determine the implicit dollar value we are placing on "Red" and "Mercedes."

In summary, I don't believe that we should pursue the topic of the summer conference with the goal of finding a simple-minded metric such as IQ or robot versions of GREs. Nor, intuitively, do I have faith here in something that assumes situation independence, linearity, and so on.

The Power of Mixes

Once we recognize that intelligence is a multifaceted concept, and that society places a high value on all aspects of intelligence, broadly construed, then we are also ready to recognize the value of healthy *mixes*. *Instead of optimizing the average "IQ" of a robot community, we should instead seek to "optimize" the effectiveness of the community*—perhaps omitting those items we expect or want humans to continue to do. Moreover, in "optimizing" we should apply nonlinear schemes that assure that we don't end up with able mediocrities. For example, in human society most of us believe that we benefit from having at least some people who are extremely good at mathematics, physics, written verbal matters, spoken language, the arts, and even the difficult human skills associated with the very best of leaders on the one hand, or the best of clinical psychologists on the other. But we don't require all of these skills from everyone.

To use a different analogy, consider how we go about dealing with medical issues. Perhaps some readers have a single physician who "does everything" from delivering babies to extracting brain tumors, but the rest of us seek to have a mixture that includes top diagnosticians (the best of whom are very smart in the traditional sense), very good internists who deal more with quantity than the with the hardest cases, and various and sundry specialists. Some of the specialists may be superb at some skills (e.g., microsurgery), but pretty poor at others. Whether this is an urban myth or reality, I don't know, but I believe that it is widely accepted that surgeons are not uncommonly a bit blockheaded and lacking in both subtleties and ability to read and care for human beings except in "mechanical" ways. Many surgeons even kid about this, describing themselves as world-class plumbers. Now, suppose that we wanted to choose a mix of doctors for a community on the moon. Would we look for some metric, test everyone, and then optimize, or would we instead identify many attributes and assure that all were adequately represented?

The Fundamental Challenge of Defining and Measuring Wisdom and Common Sense

Despite familiarity with the hilarious (or infuriating) shortcomings of some "artificial intelligence" programs, I am not particularly mystical about issues such as wisdom and common sense. Intuitively, I believe that they have to some extent been over-rated as a reaction to failures of the straightforward rule-based approaches in AI. I suspect that with large enough computers and sufficient emphasis on and time spent in training with neural nets and other technologies, machines will eventually have remarkably good skills that include what look like wisdom and common sense.

Nonetheless, this remains a frontier area for research. Measurements would depend not just on the intelligence “wired in,” but the intelligence developed by experience and the data bases provided initially and built up over time. As we know from discussions in many forums, it is notoriously difficult at present to measure the information, knowledge, or value in data bases. This, then, is just a warning of a different type.

Are We Talking About Intelligence, Humanity, or What?

We may be erring in focusing too exclusively on “intelligence,” given that the term “intelligence” is usually associated with matters distinct from ethics, morality, or spiritualism (broadly construed). It is of interest to note that this mistake was *not* made by the late, great Isaac Asimov. It was not accidental that Asimov, rather than more pedestrian writers, took on these issues directly.

As a hypothesis, it seems to me that *it will continue to prove impossible to achieve top-notch “intelligent performance” across a wide range of situations without having principles that look more like ethics than electrical engineering*. We know that one of the special characteristics of intelligent people is that they learn, taking on knowledge and skills that go beyond what they were “programmed for.” However, without some kind of principles to act as filters, what machines (or, for that matter, people) choose to learn and experiment with may prove dangerous. Again, we can look to science fiction for examples.

Assuring Exploration and Mutation

Although we may differ among ourselves about the meaning and existence of “progress,” most of us would agree that the processes of evolution such as mutation and natural selection have profound effects. Suppose that there were no mutations, or that there were no means by which to select. What might then have happened? In a sense, we know. For example, we know of the extreme vulnerability of populations when they encounter a disease that is new to them. And we know of the extreme vulnerability of overly “nice” communities when they become prey to “bad guys.” What implications does this have for defining and measuring machine intelligence? Well, the answer would differ if we had in mind only specialists such as window washers, rather than colonizers of some hostile planet. However, for some purposes at least, I would think that what we would seek to define and measure—perhaps under the rubric of a generalized notion of “intelligence—would include attributes such as audacity, curiosity, and the ability to “mutate” (in a sense to be defined).

Some Technical Issues in Thinking About Building Intelligent Models

Much has been written about artificial intelligence modeling. I would add here only a few observations based on personal experience. Some of this involved building a massive analytic war gaming system during the cold war, one in which we had Red, Blue, and Green agents representing the Soviet Union/Warsaw Pact, United States/NATO, and various third countries. These agents made decisions about war, strategy, escalation, deescalation, and termination amidst the events generated by a simulation.

The first observation is that such models are arguably likely to be more useful if they reflect a strong design rather than, e.g., a more unstructured approach such as lots of miscellaneous rules and an inference engine. Even if performance in particular tasks might be very high with the latter approach, credibility and understandability tend to go with structure and with the ability to trace rationales.

Machines will need models of other machines, and highly simplified models of the other machines' modeling. There is no infinite recursion here because—if for no other reason—uncertainties in key inputs to judgments are sufficiently large that fine-tuning doesn't work well. In our work, Blue's decisions were based on a model of Red, which in turn had a highly simplified model of Blue. Both Red and Blue could learn to some degree as the simulation proceeded, although this was wired-in learning such as changing planning factors based on events in the simulation and assessing which opponent model seemed best given observed behavior.

The second observation is that multiresolution modeling (MRM) is extremely important in such work (and in other types of modeling as well). By MRM I mean modeling that provides alternative levels at which to make inputs, as distinct from modeling that merely provides intermediate- and highly aggregated displays, but that does all calculations from the lowest level upward.⁶

MRM is important for many reasons, but one of them is relevant here. Higher level intelligent behavior depends on higher-level models, not on calculations from incredible depth. The reasons relate to the enormous uncertainties that exist at lower levels (higher resolution)—not only in “data,” but also in algorithms. This is part of the celebrated “bounded rationality” problem explained by Herbert Simon. As a result, real people (and at least some intelligent models) must be able to reason and decide at the level of abstractions. Abstractions often get built into models willy-nilly, but there is great benefit in designing them in from the start. Ideally, models would also be able to infer their own abstractions on the basis of experience. That is surely plausible with newer technology, but we've got a ways to go, to say the least. In the meantime, good design can be quite helpful. I believe that one of the best ways to “measure” the intelligence of machines will probably be to review the hierarchical concepts it uses and the processes used to move up and down those hierarchies. That is, just as we assess unintelligent computer programs not only in terms of sampled behavior, but also in terms of inputs, structure, etc., so also for intelligence.

Some of this has interesting linkages to common sense, understandability, cause-effect relationships, and learning. As a rule of thumb, I believe that a model intended to work at level n of resolution should be accompanied by models at levels $n+1$ and $n-1$. The more abstract version may be needed for planning functions such as screening, and the more detailed version may be needed to provide “explanations” (a highly relative concept) and the potential for a kind of learning that would adjust the level- n model. Experiences that may appear magical at the intended level, level n , may be explainable at level $n+1$ of resolution and it may be possible to use the experiences to recalibrate lower-level assumptions and generate new abstractions. However, if the more detailed model doesn't even exist, then it would seem that the only recourse would be for the machine to use various and sundry techniques such as statistical analysis to infer what are additional variables. There are severe shortcomings to such an approach—if, at least, it is feasible to do better. This said, it is clear that humans do have the capability—with considerable effort—to see new things and find new ways to reason without

⁶ One paper on the subject is Davis and Bigelow, *Experiments in Multiresolution Modeling*, RAND, 1998, which is available online at www.rand.org/personal/pdavis. Alex Mystel has also written on related subjects for some years. Bernie Zeigler discusses related matters such as morphism in his text (*Theory of Modeling and Simulation*, 2000).

them having been wired in our software. But we all know how useful it is to have analogies, metaphors, or theories to help.

It follows that one measure of intelligence might be the structural richness of reasoning models: is it sufficient to accommodate a good deal of experience-based learning?

Exploratory Analysis as a Key to Measuring Intelligence in an Uncertain World

One of the principles of our discussion of intelligence should be that the intelligence of a machine cannot usefully be judged independent of context. “Performance” measures exist, of course (e.g., processing speed), but *how “intelligent” something is needs to be measured in relationship to both tasks to be done and contexts in which to do them.* These, of course, are extremely uncertain. This is obvious enough, but by analogy with my work in policy analysis I would argue that special methods are needed to make use of this notion. In particular, we should plan to construct what I have variously called “scenario spaces” or “assumptions spaces” in which to test our behaviors. Not only is it insufficient to pick an allegedly representative context, and work away at measurements for that, it is also not sufficient to do sensitivities around that context. Key reasons are as follows. First, there may not be a meaningful best-estimate or representative context; instead, there may be massive uncertainties that make any of many possible and very different contexts plausible. Second, the effects of contextual variables may be highly interactive, so that any linear approach to sensitivity testing would fail.

The approach my colleagues and I have used in this regard involves “exploratory analysis,” which emphasizes studying the problem (e.g., assessing behavior’s effectiveness) in a vast scenario space that is designed for comprehensiveness rather than detail. I refer to both parametric and probabilistic explorations. In the first, one discretizes the context’s defining variables, and creates experimental designs that consider all (or a cleverly sampled subset) of the many combinations. In simple cases, we can do the full factorial design. In the second approach, one represents the defining variables’ uncertainty with uncertainty distributions. Ultimately—after initial exploration—one settles on a hybrid approach in which some key variables are parameterized (so that one can see cause-effect relationships in output displays) and the others are treated probabilistically and convolved.

It is possible in such exploratory-analysis work to gain insights about a system’s effectiveness over an enormous range of conditions (and with different measures). Once that is done, one may also want to delve into details—perhaps far into the morass of details—but at least one will know where the potential paydirt is.⁷

Fortunately, recent technology makes a great deal of this type of thing feasible—even with PCs on our desktop at home. We are already at the stage where much can be learned by “flying through the space of outcomes” using clever graphics, and thereby seeing what regions (what combination of variable values) are most important (e.g., acceptable or unacceptable outcomes).

These exploratory analysis methods could prove quite powerful in the task of assessing the intelligence of machines.

⁷ Exploratory analysis is discussed at some length in a forthcoming monograph, “Exploratory Analysis of Strategy Problems Amidst Massive Uncertainty,” by me and Richard Hillestad. A short paper on the subject is available upon request.

Hierarchic Social Entropy and Behavioral Difference: New Measures of Robot Group Diversity*

Tucker Balch
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3891

Abstract

As research expands in multiagent intelligent systems, investigators need new tools for evaluating the artificial societies they study. It is impossible, for example, to correlate heterogeneity with performance in multiagent robotics without a quantitative metric of diversity. Currently diversity is evaluated on a bipolar scale with systems classified as either heterogeneous or homogeneous, depending on whether any of the agents differ. Unfortunately, this labeling doesn't tell us much about the extent of diversity in heterogeneous teams. How can it be determined if one system is more or less diverse than another? Heterogeneity must be evaluated on a continuous scale to enable substantive comparisons between systems. To enable these types of comparisons, we introduce: (1) a continuous measure of robot behavioral difference, and (2) hierarchic social entropy, an application of Shannon's information entropy metric to robotic groups that provides a continuous, quantitative measure of robot team diversity. The metric captures important components of the meaning of diversity, including the number and size of behavioral groups in a society and the extent to which agents differ. The utility of the metrics is demonstrated in the experimental evaluation of multirobot soccer and multi-robot foraging teams.

1 Introduction

Heterogeneous systems are a growing focus of robotics research [FM97, GM97, Par94, Bal99]. Presently, diversity in these systems is evaluated on a bipolar scale; systems are classified as either *heterogeneous* or *homogeneous* depending on whether any of the agents differ. This view is limiting because it does not permit a quantitative comparison of heterogeneous systems. A principled study of diversity

requires a quantitative metric. Such a metric would enable the investigation of issues like the impact of diversity on performance, and conversely, the impact of other task factors on diversity. To address this, we propose *social entropy* (computed using Shannon's information entropy formulation [Sha49]) as an appropriate measure of diversity in robot teams.

In this paper we briefly introduce the mathematical formulation of individual robot difference and robot societal diversity. More details and examples are provided in [Bal00].

2 The meaning of diversity

What does *diverse* mean? Webster [MW89] provides the following definition:

di.verse *adj* **1:** differing from one another: unlike **2:** composed of distinct or unlike elements or qualities

Clearly, difference plays a key role in the meaning of diversity. In fact, an important challenge in evaluating robot societal diversity is determining whether agents are alike or unlike. Assume for now that any two agents are either alike or not.

Now consider what *diverse* means for societies composed of several distinct subsets. To make the discussion more concrete, suppose the "society" under examination is a collection of four different shapes: circles, squares, triangles and stars. Figures 1 and 2 illustrate several sets of shapes as examples of ways the groupings can differ. The goal is to develop a quantitative metric that captures the meaning of diversity illustrated in these examples.

First, how should the number of distinct subsets in a society impact the measured diversity? Consider Figure 1: four sets of 12 shapes. Each set has a different number of homogeneous subsets; from one homogeneous subset in Figure 1a (all circles) to four in Figure 1d. This example suggests that the number of homogeneous subsets in a society is an important component of measured diversity.

*This is an abbreviated version an article published in *Autonomous Robots*, vol 8, no 3.

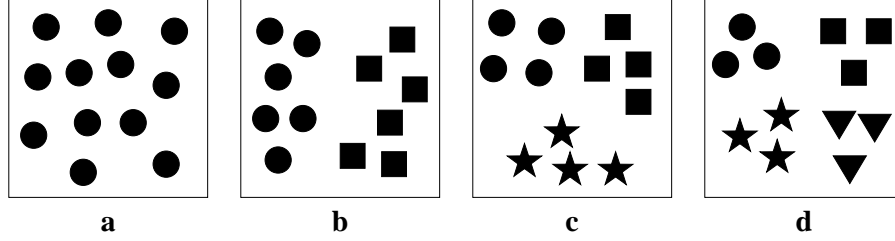


Figure 1. Several collections of shapes. The number of homogeneous subsets in each collection grows from one in a to four in d.

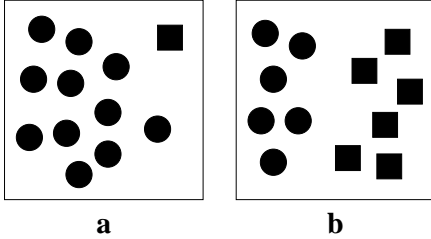


Figure 2. In both of these groups there are the same number of shapes and the same number of homogeneous subsets, but the proportion of elements in each subset is different.

Now consider Figure 2. Which group of shapes is more diverse? In both cases there are exactly 12 shapes and exactly two different types. In Figure 2a however, there is a much higher proportion of circles than in 2b where there is an equal number of circles and squares. This example suggests that the relative proportion of elements in each subset is an important component of diversity.

These examples highlight the fact that the *distribution* of the agents between homogeneous subsets is at the core of the meaning of diversity. In light of this observation, we make the following commitment: the measured diversity of a multiagent society depends on the number of homogeneous subsets it contains and the proportion of agents in each subset.

3 Simple social entropy

How should diversity be quantified? The properties Shannon sought in a measure of information uncertainty are also useful in the measurement of societal diversity [Sha49]. In fact, researchers in a number of disciplines have adopted information theoretic concepts of diversity. Information entropy is used by ecologists as a means of evaluating species' diversity [LVW83, LW80, Mag88], by so-

ciologists as a model of societal evolution [Bai90], and by taxonomists as a tool for evaluating classification methodologies [SS73, JS71].

Before proceeding we must introduce some notation:

- \mathcal{R} is a society of N agents with $\mathcal{R} = \{r_1, r_2, r_3 \dots r_N\}$
- \mathcal{C} is a classification of \mathcal{R} into M possibly overlapping subsets.
- c_i is an individual subset of \mathcal{C} with $\mathcal{C} = \{c_1, c_2, c_3 \dots c_M\}$
- $p_i = \frac{|c_i|}{\sum_{j=1}^M |c_j|}$ is the proportion of agents in the i th subset; and $\sum p_i = 1$.

In the last section we argued that the measured diversity of a system should reflect the number of groups in the system and the distribution of elements into those groups; diversity should therefore be a function of M and the p_i s as defined above. Assume that a diversity metric exists and call it H . The diversity of a society partitioned into M homogeneous subsets is written $H(p_1, p_2, p_3, \dots, p_M)$. So, for instance, the diversity of the group of shapes depicted in Figure 2a is $H(\frac{1}{12}, \frac{1}{12})$, while the diversity for the group of shapes in Figure 2b is $H(\frac{1}{2}, \frac{1}{2})$. The diversity of a particular robot society \mathcal{R}_a can also be expressed $H(\mathcal{R}_a)$.

Shannon prescribed three properties for a measure of information uncertainty [Sha49]. With slight changes in notation, they are equally appropriate for a measure of societal diversity:

Property 1 continuous: H should be continuous in the p_i .

Property 2 monotonic: If all the p_i are equal, (i.e. $p_i = \frac{1}{M}$), then H should be a monotonically increasing function of M . In other words, if there are an equal number of agents in each subset, more subsets implies greater diversity.

Property 3 recursive: If a multiagent society is defined as the combination of several disjoint sub-societies, H for the new society should be the weighted sum of the individual values of H for the subsets. This property is important for the analysis of recursively composed societies (e.g. [MAC97]).

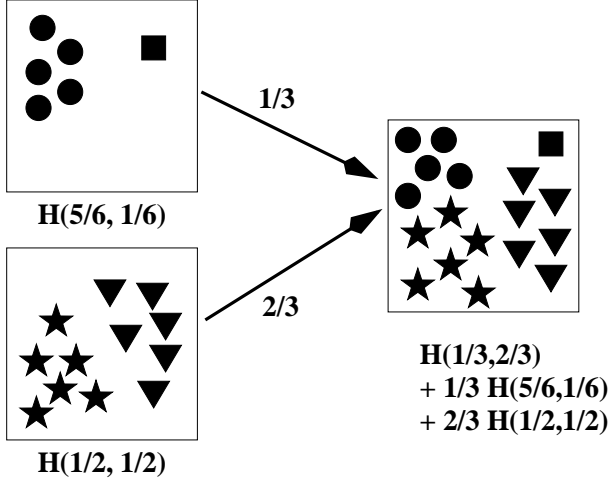


Figure 3. A new society (right) is generated by combining two others (left). The diversity of the new society is a weighted sum of the individual values of H for the subsets.

The meaning of the requirement that H be recursive is illustrated in Figure 3. The two groups on the left are combined into a new society on the right. In general, for a society \mathcal{R}_c composed of two societies, \mathcal{R}_a and \mathcal{R}_b , the recursive criteria ensures that:

$$H(\mathcal{R}_c) = H(\alpha, \beta) + \alpha H(\mathcal{R}_a) + \beta H(\mathcal{R}_b)$$

where α is the proportion of agents in \mathcal{R}_a , β is the proportion of agents in \mathcal{R}_b and $\alpha + \beta = 1$.

Shannon's *information entropy* meets all three criteria [Sha49]. The information entropy of a random system X is given as¹:

$$H(X) = -K \sum_{i=1}^M p_i \log_2(p_i) \quad (1)$$

where K is a positive constant. Because K merely amounts to the choice of a unit of measure, Shannon sets $K = 1$ [Sha49]. Equation 1 (with $K = 1$) is adopted for the measurement of multiagent societal diversity. $H(\mathcal{R}_a)$ is the *simple social entropy* of agent society \mathcal{R}_a .

¹ $H(X)$ is used in coding theory as a lower-bound on the average number of bits required per symbol to send multi-symbol messages. The random variable X assumes discrete values in the set $\{x_1, x_2, x_3 \dots x_M\}$ (the alphabet to be encoded) and p_i represents the probability that $\{X = x_i\}$.

In addition to Properties 1, 2 and 3, H has a number of additional properties that further substantiate it as an appropriate measure of diversity. First, as we would expect, H is minimized for homogeneous societies; these groups are the least diverse. Also, for heterogeneous groups H is maximized when there are an equal number of agents in each subset. More precisely:

Property 4: $H = 0$ if and only if all the p_i but one are zero. In other words H is minimized when the system is homogeneous. Otherwise H is positive.

Property 5: For a given M (number of homogeneous subsets), H is maximized when all the p_i are equal, i.e. $p_i = \frac{1}{M}$. This is the case when there are an equal number of agents in each subset.

Property 6: Any change toward equalization of the proportions p_1, p_2, \dots, p_M increases H . Thus if $p_1 < p_2$ and we increase p_1 , decreasing p_2 an equal amount so that they are more nearly equal, H increases. An important implication is that there are no locally isolated maxima.

Even if these properties are desirable in a diversity metric, why choose information entropy over another function possessing the same properties? Because, as it turns out, information entropy (Equation 1) is the *only* function satisfying Properties 1, 2 and 3. Shannon proved this result using the mathematically equivalent properties he required of an information uncertainty metric [Sha49].

The entropy of a number of example systems using this metric is given in Figure 4.

4 Classification and clustering

The discussion of diversity left open the question of how agents are classified into subsets. It was assumed that any two agents are either alike (in the same subset) or unlike. In actuality, the robotic agents to be classified are distributed in a multi-dimensional space where the dimensions correspond to components of behavior and difference corresponds to the distance between agents in the space. Difference between agents is likely to vary along a continuous spectrum instead of in the binary manner assumed previously.

The challenge of finding and characterizing clusters of elements distributed in a continuous multi-dimensional space is exactly the problem faced by biologists in building and using taxonomic systems. In the case of biology the dimensions of the space represent aspects of morphology or behavior that distinguish one organism from another. In this research the dimensions are the components of behavior that distinguish one robot from another.

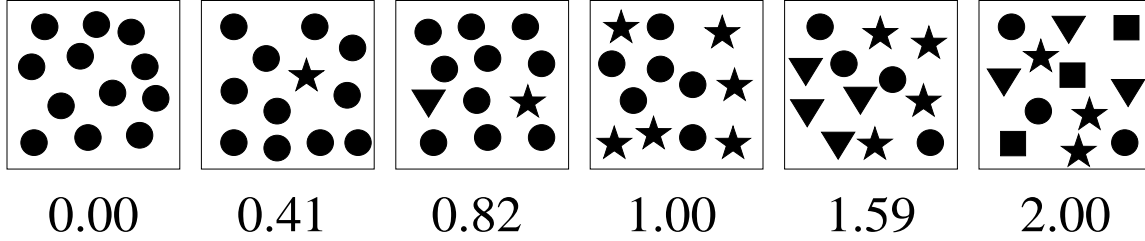


Figure 4. A spectrum of diversity. In the diagram above, each of the six squares encloses a multiagent system, from least diverse (homogeneous) on the left, to most diverse (most heterogeneous) on the right. The *simple social entropy*, a qualitative measure of diversity, is listed underneath each system.

The aims of taxonomic classification are distinct from other types of classification in that one goal is to arrange the elements in a hierarchy reflecting their distribution in the classification space. Conversely, many classification tasks only require a simple partitioning of the space (e.g. categorizing e-mail into folders). Taxonomic trees (the end result of the taxonomic classification process, e.g. Figure 5) are potentially more useful in the analysis of diversity than simple partitionings because they provide more information about the society’s spatial structure.

Biology offers a rich literature addressing this problem. In fact, an entire field — *numerical taxonomy* — is devoted to ordering organisms hierarchically using principled numerical techniques [SS73, JS71]. Many of the approaches in numerical taxonomy are directly applicable to the problem of robot classification. They include mechanisms for building and analyzing classification structures (e.g. taxonomic trees) and for identifying organisms on the basis of these structures.

Techniques from numerical taxonomy address the problem of how to classify organisms, or groups of organisms, at various levels. At the lowest level in biological classification for instance, humans and gorillas are more likely to be classified together than, say, humans and dogs. But at a higher level, primates are in fact grouped with canines in the class *mammalia*. Dendrograms provide an orderly hierarchic view of these classifications. While dendrograms *per se* are not necessary for the evaluation of diversity, they are useful visualization tools and their construction provides clues for the evaluation of overall societal diversity.

Dendrograms are constructed using a clustering algorithm parameterized by h , the maximum difference allowed between elements in the same subset. The notation $D(a, b)$ is used to refer to the difference between the elements a and b . In most applications the difference metric is normalized so that taxonomic distance between any two elements varies between 0 and 1. When $h = 1$ all elements are grouped together in one cluster (see the cluster at the top right in Figure 5 for example). As h is reduced from 1 down to 0

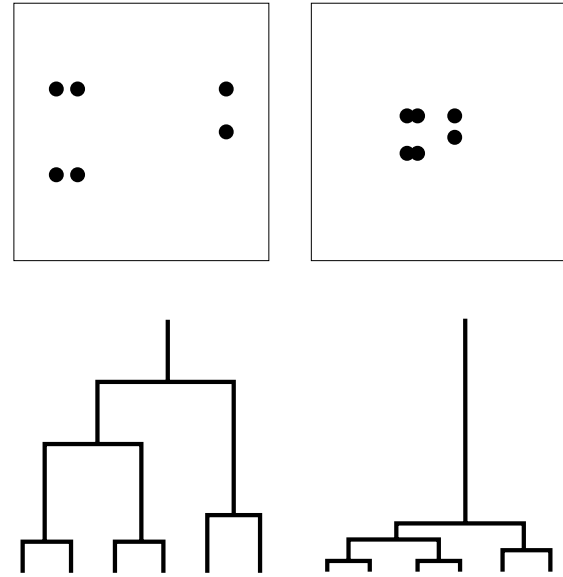


Figure 6. The branching structure of the dendrograms for these two societies is the same. However, the more compact distribution of elements in the system on the upper right is reflected in the branches being compressed towards the bottom of the corresponding dendrogram (lower right).

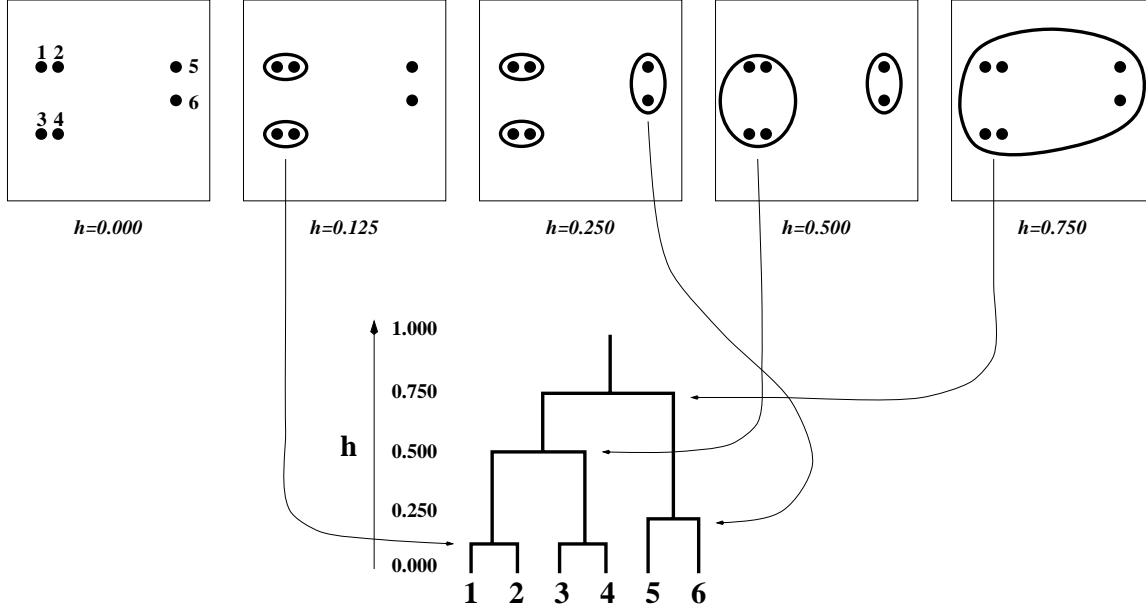


Figure 5. Example classification using numerical techniques. The top row shows how the system is clustered at several levels, parameterized by taxonomic level h (h is distinct from information entropy H). The classification is summarized in a taxonomic tree, or dendrogram (bottom). Strong similarities between elements are indicated by grouping near the bottom of the dendrogram; weaker similarities between groups are reflected in converging branches at higher levels.

cluster boundaries change; the number of subsets increases as they split into smaller clusters. The splits are reflected as branches in the dendrogram. Finally, when $h = 0$ each element is a separate cluster; a “leaf” at the bottom of the dendrogram “tree.”

Dendrograms can reveal subtle differences in societal structure. Figure 6 for example, shows two societies with the same relative arrangement of elements, but one grouping is compact while the other is spread out over a larger area. The difference in scale is reflected in a compressed dendrogram for the spatially compact society (Figure 6 right). Can these differences be accounted for in the evaluation of diversity?

The spatial extent of elements in a taxonomic space is a reflection of the degree of difference between agents. Note that sensitivity to the degree of difference between elements in hierarchic clustering depends on h . Because h is a *parameter* of the clustering algorithm, it can be varied to examine clusterings at any scale. Hierarchic algorithms are, in effect, variable power clustering microscopes. For values of h near zero the tiniest difference between elements will cause them to be classified separately, while the clusterings at large values of h reveal societal structure at a macroscopic level. This feature is exploited in the development of a diversity measure sensitive to differences in the spatial

size of societies.

5 Hierarchic social entropy

Now consider how these tools from numerical taxonomy can be applied to the measurement of diversity. The discussion of hierarchic clustering algorithms above described how the number and size of clusters depend on h . But how is simple social entropy impacted by changes in h ? Since the partitioning of a society is based on h the entropy also depends on it. An example of the relationship is illustrated in Figure 7. Entropy changes in discrete steps as h increases. Note that points where change occurs correspond to branch points in the dendrogram.

Compare the dendrograms and entropy plots of the two societies in Figure 7. As in the earlier example, the two groups have the same relative structure, but the society represented on the right is more compact, resulting in branching compressed towards the bottom of the tree. The difference in scale is also readily apparent in the plots of entropy. Entropy drops to zero much more quickly in the plot corresponding to the compact society. Because the value of simple entropy depends significantly on h when hierarchic clustering is used, we augment the notation to account for this:

$$H(\mathcal{R}, h) = H(\mathcal{R}) \text{ for the clustering of } \mathcal{R} \text{ at taxonomic level } h$$

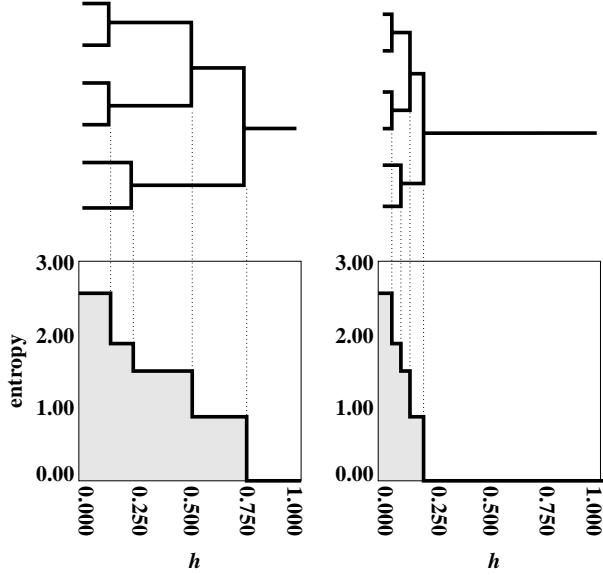


Figure 7. Entropy depends on h . A comparison of entropy versus h for two societies. For clarity, the dendrogram is rotated 90 degrees.

H is a function of \mathcal{R} and h because the classification of agents into subsets, and therefore the entropy, depends on them both. This highlights the fact that the entropy of a particular clustering is only a snapshot of the society’s diversity. A comprehensive evaluation of diversity should account for clustering at all taxonomic levels. This is easily accomplished using the area under the entropy plot as a measure of diversity. This augmented metric, called *hierarchic social entropy*, is defined as:

$$S(\mathcal{R}) = \int_0^\infty H(\mathcal{R}, h) dh \quad (3)$$

where \mathcal{R} is the robot society under evaluation, h is a parameter of the clustering algorithm indicating the maximum difference between any two agents in the same group and $H(\mathcal{R}, h)$ is the simple entropy of the society for the clustering at level h . Note that as $h \rightarrow \infty$ a point is reached where all elements are clustered in the same subset (the maximum taxonomic distance). $H(\mathcal{R}, h)$ drops to 0 at this point. In the behavioral difference measure used in this work, the maximum possible difference between elements is fixed at 1.0, so the upper limit of the integration is 1 rather than ∞ as in the general case.

Hierarchic social entropy is a continuous ratio measure; it has an absolute zero (when all elements are identical) and equal units. This enables a total ordering of societies on the basis of diversity. It also provides for quantitative results of the form “ \mathcal{R}_b is *twice* as diverse as \mathcal{R}_a .” This is a signifi-

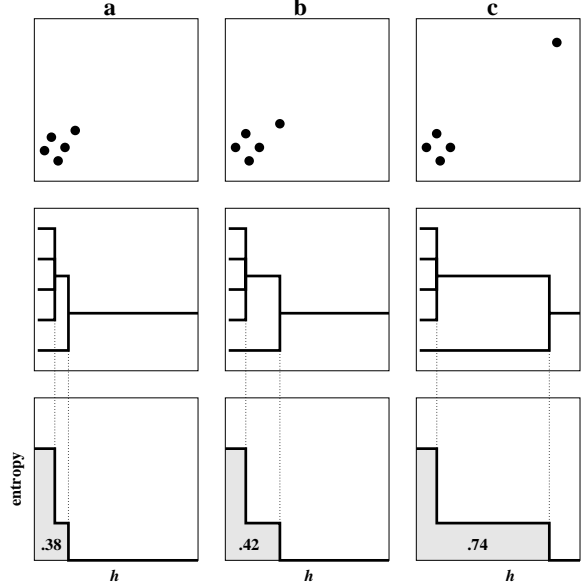


Figure 8. Hierarchic social entropy (bottom) is computed for three societies (top). The values are 0.715 for the system on the left and 1.00 for the system on the right. The calculated value increases as the element on the upper right is positioned further away from the group. Dendrograms for the groups are also displayed (middle row).

cant advantage over the categorization of systems as simply “homogeneous” or “heterogeneous.” Three example calculations of hierarchic social entropy are provided in Figure 8.

6 Behavioral difference

To summarize: hierarchic clustering is a means of dividing a society into subsets of behaviorally equivalent agents at a particular taxonomic level. Diversity is evaluated at each taxonomic level based on the number of subsets and the number of robots in each subset at that level. Integrating the diversity across all taxonomic levels produces an overall measure of diversity for the system. Previous sections have described the overall diversity metric and algorithms for clustering the agents into subsets. This section focuses on the difference metric used for clustering.

How should the behavior of two agents be compared? The technique advocated here is to look for differences in the agents’ behavioral coding. In many cases (e.g. [BBC⁺95, Mat92, GM97]) robot behavior is coded statically ahead of time, thus individuals may be directly compared by evaluating their behavioral configuration. Learning multirobot systems (e.g. [Bal97, Mat94]) pose a chal-

lenge because their behavior evolves over time. To avoid that problem in this research, the policies of learning agents are evaluated after agents converge to stable behavior.

This approach depends on three key assumptions:

Assumption 1: At the time of comparison, the robots' policies are fixed and deterministic.

Assumption 2: The robots under evaluation are substantially mechanically similar: differences in overt behavior are influenced more significantly by differences in policy than by differences in hardware.

Assumption 3: Differences in policy are correlated with differences in overt behavior.

If these conditions are not met in a particular multirobot system, the approach may not be appropriate. But the assumptions are reasonable for the conditions of this research, namely: experiments conducted on mechanically similar robots built on the same assembly line. Control systems running on the robots differ only in the data specifying each agent's policy. The comparison of these policies is the crux of the approach.

To facilitate the discussion, the following additional symbols and terms are defined:

- i is a robot's perceptual state.
- a is the action (behavioral assemblage) selected by a robot's control system based on the input i .
- π_j is r_j 's policy; $a = \pi_j(i)$.
- p_j^i is the number of times r_j has encountered perceptual state i divided by the total number of times all states have been encountered. Experimentally, p_j^i is computed *post facto*.

The approach is to evaluate behavioral difference by comparing the robots' policies. The two foraging robots introduced earlier, for example, exhibit behavioral differences that are reflected in and caused by their differing policies. In the terminology introduced above, i represents the perceptual features an agent uses to selectively activate behaviors.

Definition 1: r_a and r_b , are **absolutely behaviorally equivalent** iff they select the same behavior in every perceptual state.

In complex systems with perhaps thousands of states and hundreds of actions it may also be useful to provide a scale of equivalence. This would allow substantially similar agents to be grouped in the same cluster even though they differ by a small amount. The approach is to compare two robots, r_a and r_b , by integrating the differences between their responses, $|\pi_a(i) - \pi_b(i)|$ over all perceptual states i . If the action is a single-dimension scalar, as in a motor current for instance, the difference can be taken directly. However, complex actions like *wander* and *acquire* are treated as nominal values with response difference defined as 0 when $\pi_a(i) = \pi_b(i)$ and 1 otherwise. This approach is often used in classification applications to quantify difference between

nominal variables (e.g. eye color, presence or absence of a tail, etc.). Using this notation, a simple behavioral difference metric can be defined as:

$$D'(r_a, r_b) = \frac{1}{n} \int |\pi_a(i) - \pi_b(i)| di \quad (4)$$

or for discrete state/action spaces:

$$D'(r_a, r_b) = \frac{1}{n} \sum_i |\pi_a(i) - \pi_b(i)| \quad (5)$$

where $\frac{1}{n}$ is a normalization factor to ensure the difference ranges from 0 to 1. In the case of the discrete sum, n corresponds to the number of possible states. If r_a and r_b select identical outputs ($\pi_a(i) = \pi_b(i)$) in all perceptual states (i), then $D'(r_a, r_b) = 0$. When r_a and r_b select different outputs in all cases $D'(r_a, r_b) = 1$. In the numerical taxonomy literature, this difference is called the *mean character difference* [SS73]. The calculation parallels the idealized evaluation chamber procedure introduced earlier (Figure ??).

Equations 4 and 5 weigh differences equally across all perceptual states. This may be problematic for agents that spend large portions of their time in a small portion of the states. Consider two foraging robots that differ only in their reaction to blue attractors. If, in their environment, no blue attractors are present the agents would appear to an observer to have identical policies.

There may be other important reasons certain states are never visited. In learning a policy, for instance, the robots might discover in early trials that certain portions of the state space should be avoided due to large negative rewards. Because these portions of the space are avoided, the agents will not refine their policies there, but avoid them entirely. It is entirely possible for the agents to differ significantly in these portions of the space even though they may appear externally to behave identically.

To address this, the response differences in states most frequently visited should be emphasized while those that are infrequently experienced should be de-emphasized. This is accomplished by multiplying the response difference in each situation by the proportion of times that state was visited by each agent ($p_a^i + p_b^i$). Formally, **behavioral difference** between two robots r_a and r_b is defined as:

$$D(r_a, r_b) = \int \frac{(p_a^i + p_b^i)}{2} |\pi_a(i) - \pi_b(i)| di \quad (6)$$

or in discrete spaces

$$D(r_a, r_b) = \sum_i \frac{(p_a^i + p_b^i)}{2} |\pi_a(i) - \pi_b(i)| \quad (7)$$

When r_a and r_b select differing outputs in a given situation, the difference is normalized by the joint proportion of times they have experienced that situation.

7 Conclusion

This work is motivated by the idea that behavioral diversity should be evaluated as a *result* rather than an initial condition of multirobot experiments. Previously, researchers configured robot teams as homogeneous or heterogeneous *a priori*, then compared performance of the resulting teams [FM97, GM97, Par94]. That approach does not support the study of behavioral diversity as an emergent property in multirobot teams.

Defining behavioral diversity as an independent rather than dependent variable enables the examination of heterogeneity from an ecological point of view. How and when does diversity arise in robot teams interacting with each other and their environment? This work provides the necessary quantitative measures for this new type of investigation.

In this paper we introduce a mathematical definition of agent difference that can be used to group agents according to similarity. The grouping (or clustering) of agents is parameterized by h , a limit on how different agents can be, yet still be grouped in the same cluster. An overall diversity metric, hierarchical social entropy may then be computed using the difference metric, h , and clustering algorithms originally developed by biologists for taxonomic classification.

References

- [Bai90] K. Bailey. *Social Entropy Theory*. State University of New York Press, Albany, 1990.
- [Bal97] Tucker Balch. Learning roles: Behavioral diversity in robot teams. In *AAAI-97 Workshop on Multiagent Learning*, Providence, R.I., 1997. AAAI.
- [Bal99] T. Balch. The impact of diversity on performance in multirobot foraging. In *Proc. Autonomous Agents 99*, Seattle, WA, 1999.
- [Bal00] T. Balch. Hierarchic social entropy: An information theoretic measure of robot group diversity. *Autonomous Robots*, 8(3), July 2000. to appear.
- [BBC⁺95] T. Balch, G. Boone, T. Collins, H. Forbes, D. MacKenzie, and J. Santamaría. Io, Ganymede and Callisto - a multiagent robot trash-collecting team. *AI Magazine*, 16(2):39–51, 1995.
- [Dem92] L. Demetrius. The thermodynamics of evolution. *Physica A*, 189(3-4):417–436, November 1992.
- [FM97] M. Fontan and M. Mataric. A study of territoriality: The role of critical mass in adaptive task division. In *From Animals to Animats 4: Proceedings of the Fourth International Conference of Simulation of Adaptive Behavior*, pages 553–561. MIT Press, 1997.
- [GM97] D. Goldberg and M. Mataric. Interference as a tool for designing and evaluating multi-robot controllers. In *Proceedings, AAAI-97*, pages 637–642, July 1997.
- [JS71] N. Jardine and R. Sibson. *Mathematical Taxonomy*. John Wiley & Sons, 1971.
- [LVW83] D. Lurie, J. Valls, and J. Wagensberg. Thermodynamic approach to biomass distribution in ecological systems. *Bulletin of Mathematical Biology*, 45(5):869–872, 1983.
- [LW80] D. Lurie and J. Wagensberg. Information theory and ecological diversity. In L. Garrido, editor, *Systems Far from Equilibrium*, pages 290–303, Berlin, West Germany, 1980. Sites Conf. on Statistical Mechanics, Springer-Verlag.
- [MAC97] D. MacKenzie, R. Arkin, and J. Cameron. Multiagent mission specification and execution. *Autonomous Robots*, 4(1):29–52, 1997.
- [Mag88] A.E. Magurran. *Ecological Diversity and Its Measurement*. Princeton University Press, 1988.
- [Mat92] M. Mataric. Designing emergent behaviors: From local interactions to collective intelligence. In *Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats 2*, pages 432–441, 1992.
- [Mat94] M. Mataric. Learning to behave socially. In *Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3*, 1994.
- [MW89] Merriam-Webster. *Webster's ninth new collegiate dictionary*. Merriam-Webster, 1989.
- [Par94] Lynne E. Parker. *Heterogeneous Multi-Robot Cooperation*. PhD thesis, M.I.T. Department of Electrical Engineering and Computer Science, 1994.
- [Sha49] C. E. Shannon. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.
- [SS73] P. Sneath and R. Sokal. *Numerical Taxonomy*. W. H. Freeman and Company, San Francisco, 1973.
- [Wil92] E.O. Wilson. *The Diversity of Life*. Norton, 1992.

Developmental Performance Metrics for the Evaluation of Artificial Intelligence - A Proposal

By Dr. Anat Treister-Goren, Jack Dunietz, and Jason Hutchens*

Abstract

This paper proposes evaluation metrics for artificial intelligence that are based on two assumptions: that the Turing Test provides a sufficient subjective measure of machine intelligence, and that a behaviorist approach is necessary to achieve true artificial intelligence.

Introduction

Artificial Intelligence: definition precedes evaluation

The artificial intelligence (AI) field has strayed very far from its original interpretation by its unofficial founder, Alan Turing. Turing, who suggested a strict criterion for "intelligence", devised what came to be known as "The Turing Test", by which a computer program is said to be "intelligent" if (and only if) it "fools" a human into believing it is human. In the philosophical journal *Mind* (1950), Turing¹ posed the question "Can a Machine Think?" His answer was that if the responses from the computer were indistinguishable from those of a human, the computer could be said to be thinking.

Despite the strict criterion suggested by Turing, AI researchers diverged in multiple directions of inquiry. Today, referring to "The AI field" could mean a variety of topics including but not limited to intelligent agents, chatterbots, pattern recognition, voice recognition, machine learning or expert systems. AI applications include applications in medicine, financial investing, computer games, business, and manufacturing. Some even consider word-processing software or home appliances as AI. The field is currently in a contentious state. Even though important work has been conducted in terms of sophistication and expertise of programs, the vision that motivated the birth of the AI field

*All authors are with Ai (Artificial Intelligence NV). Contact information for Dr. Anat Treister-Goren: anat@a-i.com

is not yet fulfilled: there is neither sufficient cooperation nor agreement among its researchers.

The unfortunate result of this trend is that true advancement is inhibited. We believe, however, that a paradigm shift is inevitable. With this in mind, we propose to establish new standards and renew original concepts in an attempt to unify the field and establish evaluation standards.

In this paper we shall demonstrate that Turing's measure of artificial intelligence is indeed an appropriate method of evaluation. We show that this is particularly true when behavioristic approaches are applied to AI. Further, we maintain that a developmental approach is a necessary prerequisite for the emergence of true AI, and we show that it has proved successful in other fields. We then introduce our proposed evaluation metrics, and conclude with speculation about future progress in AI.

Renewing the definition: Turing was right.

The Turing Test

In 1950, Turing described the imitation game, nowadays referred to as the Turing Test, whereby an interrogator must determine which of two subjects is a human being, and which a computer program. Turing concluded that an inability on the part of the interrogator to reliably make a correct determination is indicative of intelligence on the part of the computer program.

The Turing Test is an appealing measure of artificial intelligence because, as Turing himself writes, "it has the advantage of drawing a fairly sharp line between the physical and the intellectual capacities of a man".

The Loebner Contest, held annually since 1991, is an instantiation of the Turing Test². In a recent thorough review of conversational systems³, Hasida and Den emphasize the absurdity of performance in the Loebner competition. They assert that a Turing test requires that systems "talk like people" and since there is currently no system to meet this requirement, ad hoc techniques make little contribution in advancing dialog technology.

We concur with Turing's methods and therefore our approach equates Artificial Intelligence with conversational skills. We further believe that engaging in a **domain-unrestricted conversation** is the most critical evidence of the existence of intelligence.

We believe that only an intelligent being can classify another as (also) intelligent and so we follow Turing's assertion that a computer program is "intelligent" if (and only if) it "fools" a human into believing he or she is conversing with another human.

Turing's Child Machine

Turing concluded his classic paper by theorizing on the design of a computer program that would be capable of passing the Turing Test. He correctly anticipated the difficulties that AI would face in the decades following his death, writing that "instead of trying to produce a program to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain".

Turing regarded language as an acquired skill, and recognized the importance of avoiding hard-wiring the computer program wherever possible. He viewed language learning in a behavioristic light, and believed that the language channel, narrow as it may be, is sufficient to transmit the necessary information, such as orders, rewards, and punishments, which the child machine requires in order to acquire language.

Turing wrote that an important feature of a learning machine is that its teacher will often be very largely ignorant of what is going on inside, although he may still be able to some extent to predict his pupil's behavior.

It is indeed unfortunate that this promising line of work was mostly abandoned by the AI community. Today we find ourselves at a crossroads - a paradigm shift is in the air. Many AI researchers are returning to the behaviorist approach that Turing suggested.

Current approaches to conversational systems

Contrary to Turing's prediction, no true conversational systems have yet been produced and none has passed an unrestricted Turing Test. The traditional approach to conversational system design has been to equate language with knowledge, and to hard-wire rules for the generation of conversations. This approach has failed to produce anything more sophisticated than domain-restricted dialog systems. Such systems lack the kind of flexibility, openness, and learning capabilities that are the essence of human conversational skills. As far as human-like conversational skills are concerned – no system has gone beyond toddler level conversation, if at all.

Since the 1950s, the field of child language research has undergone a revolution inspired by the works of Chomsky (1957)⁴ on transformational grammar on the one hand and the work of Skinner (1957)⁵ on the behaviorist theory of language on the other. Computational implementations based on the Chomskian philosophy incorporate rules for generating dialogues and conversations and have yielded disappointing results. It is our thesis that true conversational abilities are more easily obtainable via the currently neglected behaviorist approach.

Behaviorism and AI

Child language acquisition: the modern behavioristic approach

Behaviorism focuses on the observable and measurable aspects of lingual behavior. Behaviorists search for observable environmental conditions known as *stimuli* that co-occur and predict the appearance of specific verbal behavior or *responses* (Owens, 1996)⁶. This is not to say that behaviorists deny the existence of internal mechanisms, and they do recognize that studying the physiological basis is necessary for a better understanding of behavior. What behaviorists object to are internal structures or processes with no specific physical correlate that are inferred from behavior. Thus, behaviorists object to the kind of grammatical structures proposed by linguists and claim these only complicate explanations of language acquisition (Zimmerman, & Whitehurst, 1979)⁷. Their approach is functional rather than structural. They focus on the functions of language, the stimuli that evoke verbal behavior, and the consequences of language performance.

Skinner argues that psycholinguists should ignore traditional categories of linguistic units but should treat language as they would any other behavior, search for the functional units as they naturally occur, and then discover the functional relationship that predicts their occurrence. Behaviorism is focused on reinforced training since it regards language as a skill that is not essentially different from any other behavior. Speaking (and understanding speech) must be controlled by stimuli from the environment in the form of reinforcement, imitation, and successive approximations to mature performance (sometimes referred to as “shaping”). Skinner takes the extreme position that the speaker is merely a passive recipient of environmental pressures, having no active role in the process of language behavior or development.

According to behaviorists, changes in behavior are explained through the connection or association of stimuli in the environment and certain responses of the organism. The process of forming such associations is known as *classical conditioning*. For example, the word ‘milk’ is learned when the infant’s mother says ‘milk’ before or after feeding, and this word becomes associated with the primary stimulus (the milk itself) to eventually elicit a response similar to the response to the milk. Once a word or a *conditioned stimulus* (CS) elicits a *conditioned response* (CR), it can become an *unconditioned stimulus* for modifying the response to another conditioned stimulus. If the new CS ‘bottle’ frequently occurs with the word ‘milk’, it may come to elicit a response similar to that for the word ‘milk’. This way, words stimulate each other and this classical conditioning accounts for the interrelationship of words and word meanings. Classical conditioning is more often used to account for the receptive side of language acquisition.

Whereas classical conditioning accounts for the associations formed between arbitrary verbal stimuli and internal responses or reflexive behavior, *operant conditioning* is used to account for changes in voluntary, nonreflexive behavior that arise due to environmental consequences contingent upon that behavior. Operant conditioning is used to account for the productive side of language acquisition, being concerned with changes in behavior that arise from reactions to either rewards or punishment from the environment. All behavioristic accounts of language acquisition assume that children’s productive speech develops through differential *reinforcers* and *punishers* supplied by the environmental agents, in a process known as *shaping*. Children’s speech that most closely resembles adult speech is rewarded, whereas productions that are meaningless are either ignored or punished. The behaviorists believe that the course of language development is largely determined by the course of training, not maturation. Some behaviorists explain that language is processed as word-sequences or response-chains with the words themselves serving as stimulus for other successive words. These word chains are also known as *Markov models* of sentences (Mowrer, 1960)⁸. Imitation is another important factor in language acquisition because it allows a shortcut to mature behavior without the laborious shaping of each and every verbal response. It can be an exact copy of observed behavior but is not limited to being an exact one. The process of imitation itself becomes reinforcing and enables rapid learning of complex behaviors.

The time it takes children to acquire language is viewed as a consequence of the limitations of the training techniques rather than of the maturation of the child. Behaviorists do not typically credit the child with the knowledge of rules, with intentions or meanings, or with the ability to abstract important properties from the language of the environment. Rather, certain stimuli evoke and strengthen certain responses in the child. The sequence of language acquisition is determined by the most salient environmental stimuli at any point in time, and by the child’s past experience with those stimuli. The learning principle of reinforcement is therefore taken to play a major role in the process of language acquisition.

The Developmental Model

Application to AI

We maintain that a behaviorist developmental approach to language could yield breakthrough results in the creation of artificial intelligence. Programs can imitate, extract implicit rules, learn from experience, and can be instilled with a drive to constantly improve their performance. Language acquisition can be achieved through successive approximations and positive and negative feedback from the environment. Instilled with these capabilities, programs should evolve through critical developmental language acquisition milestones in order to reach adult conversational skills. Language acquisition milestones are both quantifiable and descriptive measures and systems could be evaluated using these measures, and could be assigned an age or a maturity level beside their binary assessment as 'intelligent' or 'not intelligent'.

Success in other fields

Developmental principles have enabled evaluation and treatment programs in fields formerly suffering from a lack of organizational and evaluative principles (Gleason, 1985⁹, Goren et al, 1996¹⁰). The developmental principles have been especially useful in areas bordering on the question of intelligence. Normative developmental language data enabled the establishment of diagnostic scales, evaluation criteria, and treatment programs for developmentally delayed populations. In other areas, such as schizophrenic thought disorder, in which clinicians often found themselves unable to capture the communicative problem of patients in order to assess their intelligence level or cognitive capability, let alone to decipher medication treatment effects on the patients, the developmental metrics proved a powerful tool (Goren, 1997¹¹).

It Can Be Done

Computational language acquisition

We are interested in programming a computer to acquire and use language in a way analogous to the behavioristic theory of child language acquisition. In fact, we believe that fairly general information processing mechanisms may aid the acquisition of language, by allowing a simple language model, such as the aforementioned Markov model, to bootstrap itself with higher-level structure.

Markov Modeling

Claude Shannon, the father of Information Theory, was generating quasi-English tests using Markov models in the late 1940's¹². Such models are able to predict which words are likely to follow a given finite context of words, and this prediction is based on a

statistical analysis of observed text.

Using Markov models as part of a computational language acquisition system allows us to minimize the number of assumptions we make about the language itself, and to eradicate language-specific hard-wiring of rules and knowledge.

To date, conversation systems based on this approach have been thin on the ground¹³, although the technique has been used extensively in related problems, such as speech recognition, text disambiguation, and data compression¹⁴.

Finding Higher-Level Structure

Shannon's Information Theory may be applied to the sequence of predictions made by a Markov model in order to find sequences of symbols and classes of symbols that constitute higher-level structure. For example, a character-level Markov model inferred from English text can easily segment the text into words, while a word-level Markov model inferred from English text may be used to 'discover' syntactic categories¹⁵.

This structure, once found, can be used to bootstrap the Markov model, allowing it to capture structure at even higher levels. It is our belief that combining this approach with positive and negative reinforcement is a sensible way of realizing Turing's vision of a child machine.

Proposed Evaluation Metrics

Our evaluation proposal to measure the performance of a conversational system is composed of both subjective and objective components.

Objective developmental metrics

The ability to converse is complex, continuous, and incremental in nature and thus we propose to add incremental metrics to complement the subjective impression of intelligence. Some examples of developmental parameters, which increase quantitatively with age, are:

- **Vocabulary size:** the number of different words spoken.
- **Mean length of utterance:** the mean number of words spoken per utterance.
- **Response types:** the ability to provide an appropriate sentence form with the relevant content in a given conversational context and the variety of forms used.
- **Degree of syntactic complexity:** for example, the ability to use embedding to connect between sentences and convey ideas.
- **The use of pronominal and referential forms:** the ability to use pronouns and referents appropriately and meaningfully.

Each stage of language acquisition sets the foundation for the next and growth is progressively measured.

The added value

The incremental measures provide an evaluation of progress in conversational capabilities over time. The descriptive increments enable capturing specific aspects of conversational capabilities. Moreover, they enable understanding the nature of the critical aspects that lead up to the ultimate goal: achieving a subjective judgment of being ‘intelligent’.

The challenge in creating maturational criteria is in combining the parameters into a meaningful profile or evaluation score. One might expect discrepancies in the development of the different aspects of conversational performance. For example, some systems may utter long, complex syntactic sentences, typical of a child at age 5 or above, but may lag in terms of the use of pronouns expected at that age. The weighting of the various maturation parameters is far from trivial.

The subjective component

We do not claim that the objective evaluation should take precedence over the subjective one, just as we do not judge children on the basis of objective measures alone. A subjective judgment is an important, if not determining criterion, in an overall evaluation.

The judgment of intelligence is in the eye of the beholder. Human perception of intelligence is always influenced by the **expectation level** of the judge towards the person or entity about to be judged (Obviously, intelligence in monkeys, children, or university professors will be judged differently). Adding objective metrics for evaluating maturity level will set up the right expectation level for a valid subjective judgment of intelligence.

Accordingly, we propose developmental metrics to establish a common denominator among various conversational systems, so that the expectation level from these systems will be realistic. Given that subjective impression is at the heart of the perception of intelligence, the constant feedback from the subjective evaluation to the objective one will eventually contribute to an optimal evaluation system for perceiving intelligence.

By using the developmental model, computer programs will be evaluated to have a maturity level in relation to their conversational capability. Programs could be at the level of toddlers, children, adolescents, or adults depending on their developmental assessment. This approach enables not only evaluating across programs but also evaluating the progress within a given program.

Conclusion and Future Work

We submit that a developmental approach is a prerequisite to the emergence of intelligent lingual behavior, and to the assessment thereof. This approach will help establish standards that are in line with Turing's understanding of “intelligence” and will enable evaluation across systems.

We maintain that the proposed paradigm shift in understanding the concepts of “*Artificial Intelligence*” and “*Language*” will result in the development of groundbreaking technology that will pass the Turing Test within the next 10 years.

References:

-
- ¹ Turing, A.M. (1950). *Computing machinery and intelligence* Mind, 59, 433-560.
- ² <http://www.loebner.net/Prizet/loebner-prize.html>
- ³ Hasida, K., & Den, Y. (1999). A synthetic evaluation of dialogue systems. In Yorick Wilks (Ed.), *Machine Conversations*, Massachusetts: Kluwer Academic Publishers.
- ⁴ Chomsky, N. (1957). *Syntactic Structures*. The Hague: Mouton.
- ⁵ Skinner, B. F. (1957). *Verbal behavior*. Englewood, NJ: Prentice-Hall
- ⁶ Owens, R. E. (1992). *Language Development*. Macmillan Publishing Company
- ⁷ Zimmerman, & Whitehurst, (1979). Structure and function: a comparison of two views of development of language and cognition. In G. Whitehurst & B Zimmerman (Eds.), *The functions of language and cognition*. New York: Academic Press.
- ⁸ Mowrer, O. H. (1960). *Learning theory and symbolic processes*. New York: Wiley.
- ⁹ Gleason, J.B. (1985). *The Development of Language*. Charles E. Merrill Publishing Company; Columbus.
- ¹⁰ Goren, A. (1997). *The language deficit in schizophrenia from a developmental perspective*. The Israeli Association of Speech and Hearing clinicians, The 33rd convention.
- ¹¹ Goren, A., & Tucker, G. & Ginsberg, G., M. (1996). *Language dysfunction in schizophrenia*. European Journal of Disorders of Communication, vol 31 (2) 467-482.
- ¹² Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.
- ¹³ Jason L. Hutchens. Introducing MegaHAL. In David M. W. Powers, editor, *NeMLaP3 / CoNLL98 Workshop on Human-Computer Conversation*, ACL, pages 271-274, January 1998.
- ¹⁴ Eugene Charniak. *Statistical Language Learning*. MIT Press, 1993.
- ¹⁵ G. Whitehurst and B. Zimmerman. Structure and function: A comparison of two views of development of language and cognition. In G. Whitehurst and B. Zimmerman, editors, *The Functions of Language and Cognition*. Academic Press, 1979.

General Scientific Premises of Measuring Complex Phenomena

H.M. Hubey, Professor

Computer Science

Montclair State University, Upper Montclair, NJ 07043

ABSTRACT

General scientific and logical premises lurking behind the art of measuring complex phenomena, specifically intelligence, are explored via fuzzy logic, probability theory, differential equations, thermodynamics, generalized dimensional analysis, philosophy and psychology.

KEYWORDS: *generalized dimensional analysis, path functions, fuzzy operators, fuzzy logic, thermodynamics, extensive variables, intensive variables*

0. THERMODYNAMICS, OSs, AND TURING

Thermodynamics is probably the classical and ideal example of a *system-theoretic* point of view, and one that is built on the twin concepts of *state* and *process*. Furthermore, it is probably the only link from physics to the study of living things, which are most likely the most complex things which humans will ever have to study. The physical sciences are the easy sciences; it is the life sciences that are the hard sciences.[1] Unfortunately, physical scientists work with powerful tools, and life sciences have restricted themselves to working with much less powerful tools[1].

Thermodynamics is a perfect example of a science whose development lead to the improvement of the measurement of a fundamental dimension of physics. It was not until Lord Kelvin saw some inconsistencies that the concept of an 'absolute' temperature scale was created. In measurements of things such as length, mass, or time we can easily envision the concept of 'zero'. But it is not so with temperature. Nobody knew what the lowest obtainable temperature was. In the arguments in the philosophy of science there exist *data-first* and *theory-first* schools. Here we have a case in which both are iteratively used. The problem of intelligence is most likely to follow this pattern of development. If the problem is in an area that has a well-developed theory, we must try to explain the phenomenon in terms of the developed theory. It is only when we cannot that we can start thinking about a new theory, and this requires datamining techniques.

An Operating System (OS) is a very complex object. It has been said that "I may not know what an OS is but I can recognize one, when I see one!". The same thing may be said about intelligence, (or cognitive ability or any of the other

related words such as awareness, consciousness, or autonomy, or even life.) The Artificial life newsgroup (alife) skipped trying to define life or artificial life. The only serious effort in this direction was made by Alan Turing. He essentially formalized the saying about the OS into intelligence. We may not know what 'intelligence' is but we know how to recognize one when we see one. Apparently when we talk about intelligence, we are talking about 'human kind' or 'human type' or 'human level' intelligence, or at least 'living thing' kind (type/level) of intelligence. We can say things about this without being able to define it precisely. It is precisely about this intelligence that Turing was referring to when he wrote about what is now referred to as the 'Turing Test'. He understood all the problems that involve discussions of this thing called intelligence many decades ago and offered his 'Gordian Knot' solution. Sometimes thinkers are unable to break through the boundaries of what has been created. Whitehead claims that Aristotle hindered the development of science for 2,000 years because nobody was courageous enough to break through the boundaries of the box for the sum total of all knowledge for human kind.

1. MEASUREMENT THEORY I

Normally, in the physical sciences, the possibility that an instrument may be capable of high precision while not being able of high accuracy does not occur to people. It can only occur if the instrument is broken. If the instrument is a very simple one (such as a ruler) we'd see immediately if there was something seriously (or obviously) wrong. If the instrument is a highly complex one, then there would be various self-tests. However, in the social/life sciences creation of 'instruments' is an art. It is quite possible for the instrument to be *reliable* (precise) but not *valid* (not accurate) or vice versa. For example, a psychologist might decide to create a questionnaire which he claims measures 'hostility'. The same person taking this test (the questionnaire) might obtain different scores at different times. So habituated are we to measuring things in this modern age that we scarcely give thought to the possibility that what is being represented as a number may be meaningless. That is the validity of the measurement i.e. that the measurement or metric actually measures what we intend to measure. In physical measurements there is usually no such problem. Validity also comes in different flavors such as construct-validity, criterion-related validity, and content-validity. Reliability refers to the

consistency of measurements taken using the same method on the same subject. (Please see Figure 1)

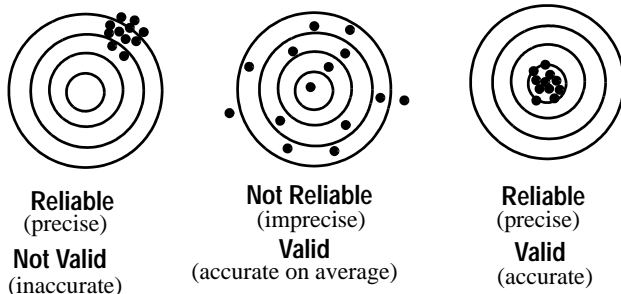


Figure 1: Reliability and Validity Analogy: One normally expects accuracy to increase with precision. However in the social sciences they are independent.

2. MEASUREMENT THEORY II

We often need to make things comparable to each other. We call this *normalization*. That is most easily done if we use numbers. For example, one way to normalize test grades is simply to divide every grade by the highest grade in class. This guarantees that the highest grade in class is 1.0. In order to be able to compare one boxing match to another a standard scoring system is used in which the same number of referees are used to score the bout, and for each round at least one boxer must be given 10 points. In Rasch measurements, we use

$$\frac{P}{1-P} = e^{\alpha - \delta} \quad (1)$$

where $P = \text{Prob}\{\text{answering correctly}\}$, α = ability, and δ = difficulty of question. However, this is not scale-free. It would probably be better to use something like

$$\frac{P}{1-P} = \frac{\alpha}{\delta} \quad \text{or} \quad \frac{P}{1-P} = 1 + \ln\left(\frac{\alpha}{\delta}\right) \quad (2)$$

In this case it is only necessary that both α and δ be measured on the same scale (somehow). Obviously, it would be best for all purposes to use numbers in the standard interval $[0,1]$.

3. MEASUREMENT THEORY III

Before we try to normalize quantities we should know what kinds of measurements we have. They determine if we can multiply those numbers, add them, or can merely rank them etc. Accordingly measurements are classified as: (i) Ratio scale, (ii) Interval scale, (iii) Ordinal scale, or (iv) Nominal scale.

Absolute (Ratio) Scale: The highest level of measurement scale is that of ratio scale. A ratio scale requires an absolute or nonarbitrary zero, and on such a scale we can multiply (and divide) numbers knowing that the result is meaningful.

Interval Scale: The Fahrenheit and Celsius scales are interval scales. The differences on these scales are meaningful but ratios are not. That is what Kelvin found out, and that is what

the absolute temperature scale is about. When measuring things such as intelligence, consciousness, awareness, or even autonomy, or hostility, we have no guarantee that we are measuring any of these on an absolute scale. There must be some other guidelines. One of the guidelines is obviously the study of various scales. In the intelligence game, psychologists have mainly relied on the *central limit theorem* in ‘hoping’ that intelligence is a result of many many different things adding up to create a Gaussian density. Thus they have contrived to make sure that test results are Gaussian.

Ordinal Scale: The next level on the measurement scale is the ordinal scale, a scale in which things can simply be ranked according to some numbers but the differences of these numbers are not valid. In the ordinal scale we can make judgements such as $A > B$. Therefore if $A > B$ and $B > C$, then we can conclude that $A > C$. In the ordinal scale there is no information about the magnitude of the differences between elements. It is possible to obtain an ordinal scale from questionnaires. One of the most common, if not the most common is the multiple-choice test, called the *Likert scale*, which has the choices: extremely likely/agreeable, likely/agreeable, neutral, unlikely/disagreeable, and extremely/very unlikely/disagreeable.

Nominal Scale: The lowest level of measurement and the simplest in science is that of *classification* or *categorization*. In categorization we attempt to sort elements into categories with respect to a particular attribute. It ranks so low on the scale that it was added to the measurement scales later. Even an animal that can tell food from nonfood can be said to have learned or can be said to know about set operations instinctively.

The most basic and fundamental idea underlying these scales which is not even mentioned, and which is extremely important for measurement of complex phenomena in the life sciences, is that in the final analysis, it is the human sensory organs that are the beginnings of all measurement. In the measurement of temperature, although a difference scale was easy to set up via the human sensory organs (and induction), it took theory and scientists to obtain an absolute scale for temperature. To obtain a difference scale the only thing necessary was for humans to note that the liquid in the glass went up when it was hotter. There was no way to know which was more hot and which less hot except via our naked senses.

This is/was as basic as knowing the difference between which of two sticks is longer than the other or which of two weights is the heavier one. Similarly in the measurement of intelligence, the final arbiter is still the naked human senses. Humans must make up the tests and decide which is more intelligent, say a chimpanzee or a dog. There can be no other way to proceed. The genius of Turing was that he realized this immediately. Therefore, Turing’s basic intuition is correct. We might not know what intelligence is but we can recognize it when we see it. Secondly, we should probably turn to nature to find examples and a hierarchy or scaling of intelligences. It would not be off the mark to accept that all living things are intelligent to a degree, and that EI (Encephalization Index) is ba-

sically a good scale on which to compare the intelligences of at least some living organisms.[2]

4. MEASUREMENT THEORY IV

Before we can even think about whether our measurements are on an absolute or difference scale we have to make sure that the objects that we deal with are *quantifiable* in some way and that we can measure them (with numbers naturally). Our handle on the problem is that the things we measure in physics (and hence engineering) come in *fundamental dimensions*. For example, dimensions of that particular branch of physics called mechanics consists of M {mass}, L {length}, and T {time}. For electrical phenomena we need one more dimension, Q (charge), and for thermal phenomena we need θ (temperature).

Then we can entertain the thought of using *dimensional analysis* for complex phenomena which is a method of reducing the number and complexity of experimental variables which affect a given physical phenomenon, using a sort of compacting technique. If a phenomenon depends upon n dimensional variables, dimensional analysis will reduce the problem to only k dimensionless variables, where the reduction $n - k = 1, 2, 3$ or 4 depending on the problem. Since these new dimensions are products/ratios of the old variables to various powers, the new dimensionless space has nonlinearly twisted and compacted the old problem in a way in which we can see regularity.

These ideas have been put to good use in biology [3]. For example, the mass of an animal grows proportional to L^3 but its surface area is only proportional to L^2 . Thus, as animals get larger they have to have larger cross-sections of bones to support all that weight. So an elephant does not look just like a large sheep. These ideas have to be taken into account when prototypes, say, airplanes are tested in wind tunnels. Many other things having to do with scaling of living things such as metabolism, oxygen consumption, heat exhaustion, cooling etc. can be found in Schmidt-Nielsen[3]. For example, one way to make different animals's brains comparable is to compare not their brain capacities but the ratio of their brain mass, b , to their body mass B . Until recently, there was no method that could cluster the variables in similar ways as above so that nonlinear dimensional compaction was not available, but now there is a generalized data-driven method.[4]

5. PHILOSOPHY

Why do we do philosophy? One reason is because we do not want to 're-invent the wheel'. If philosophers have already thought about this topic, we should at least be aware that thought has been expended and results have been achieved.

Operationalism: The problem of what is being measured in quantum mechanics was solved during the early part of this century by 'operationalism' an idea (by Bridgeman) that

the operations that are being executed define what is being measured. As long as everyone does the same thing, we are guaranteed that we all measure the same thing. In the measurement of something like intelligence, obviously, the problem of validity remains.

Quality vs Quantity: Thermodynamics, gave us the concept of *extensive* and *intensive* variables. It is often remarked in narratives that a fundamental difference exists which can be characterized by the words 'quantitative' vs. 'qualitative'. Often what is meant by the word qualitative is "intensive" since concepts often characterized as a quality can also be quantified. If a system consisting of a lot of 10,000 TVs is split into two sets at random, the quality of the two subsystems will equal each other and the quality of the TVs of the whole original system. A state of a system is characterized by a set of parameters. If we split a thermodynamic system (say a container of gas) in half some of the parameters will obey $X_1 + X_2 = X_s$ and others will obey $x_1 = x_2 = x_s$. The former (upper case) are *extensive* parameters, and the latter *intensive* parameters.

Open vs Closed: The concepts open vs closed (*endogenous vs exogeneous*) are obviously very closely related to each other. In a closed system there can be no such thing as an exogeneous variable. At the same time, in general there is really no accurate or clear definition of what an open system is. In thermodynamics from where these ideas are probably borrowed, an open system is one which exchanges mass with its surroundings. A closed system may exchange heat, and do work on its surroundings, or have work done on it by its surroundings. Additionally, heat and work are processes. In other words, they are not *point functions*, but *path functions*.

In general in mathematical modeling via differential equations, the surroundings (*forcing* or *source* term) is everything that does not have the system variable in it and usually put on the rhs. However, when these concepts are specifically applied to intelligence, we have to clarify what it is that the system exchanges with its surroundings. The concept can apply to both exchanging data and or information with its surroundings. At the same time, the word "open" may be used to refer only to the problem at hand (i.e. if the problem is "open-ended"), but then it is not about generalized intelligence but about a specific problem. To generalize it we will then be forced to think about what little we know about how the brain does its work or how to generalize from the mathematical methodology that presently exists (i.e. logic, probability theory, etc). [1]

Many-as-One: The most fundamental such concept according to modern math is 'set' and forms the basis of logic, where philosophers are at home. This idea is the building block of all systems. A body is not just a parts list although it is comprised of many subsystems thus is not merely a set. We have many ways in mathematics of treating many things as one. A *tensor* is a general object of any degree. A zero dimensional

tensor is a *scalar*. A one dimensional tensor is a *vector* or an *n-tuple*. A two dimensional tensor is called a *matrix*. In addition to this, from computer science we have the latest, and more flexible concept of hierarchical ordering via OOP (object-oriented programming) in which an object is a set of parameters without necessarily being merely a set or a vector.

Parallel vs Serial (sequential): This is one idea that occurs quite often. Some problems are parallelizable. For example, to dig a large ditch if we hire 100 workers as long as they do not interfere with each other, the ditch-digging will go at a rate 100 times as fast as before. However, if I want to send a message with a messenger, it does not matter if we use 100 messengers. The increase in the number of messengers might increase the *reliability* but will not affect the speed of the delivery. But parallelity also has to do with simultaneity (not always in time), choices, and substitutability, and logic.[7]

Trade-offs and Logic: We can sometimes trade-off something for something else in which case these things are substitutes of some kind. This idea shows up in logic as a logical-OR (co-norm). In the psychology and cognitive science literature, many different components of intelligence are posited. It is quite possible that some of these intelligences are composed of other more primitive types. If so, then are some of these substitutes for each other?

6. PSYCHOLOGY & COGNITIVE SCIENCE

Obviously throughout most of the century those who have worked on the nature and measurement of intelligence (almost always human intelligence) have been psychologists. They have had recourse to and benefited from methods and argumentation in both philosophy and physics. The kinds of questions with which they have toiled can be summarized in modern (and mathematical) terms as:

i) What kind of a quantity is intelligence? Is it *binary* or measurable on some scale? What kind of a scale is appropriate? Is it an *ordinal*, *interval*, or an *absolute* (ratio) scale?

ii) Is it an *additive function* of its constituents, the most important ones for purposes of simplification being hereditary (nature) and environmental (nurture)? Or is it a *multiplicative function*? Is it logarithmic function, an exponential function or a polynomial function of its variables?

iii) Is it a *vector/tensor* or a *scalar* (Spearman's g)? In other words, can a single number be produced from many numbers which is meaningful? Is there a hierarchy of intelligences, some of which subsume some of the others?

iv) Is it a *state* or a *process* ? In other words is it a *point function*, or a *path function*? Is it a *quality* or a *quantity*? In other words, is it an *extensive* variable or an *intensive* variable?

v) The *nature vs nurture* problem: Are the differences in intelligence among humans due mostly to heredity or environment?

There is a related (and incorrectly stated) version of (v) which is "Is intelligence mostly genetic?" The answer is quite plainly that intelligence is mostly genetic if intelligence is discussed in its most general form, that is including machine intelligence and animal intelligence. However the answer to (v) is much more complicated.[5]

An almost perfect example of a vector of cognitive science is color. We all know what colors are but they would be virtually impossible to explain to someone who was congenitally blind. If we did attempt to "explain" colors by explaining that "black is the absence of color and white is a mixture of all the colors" it is likely that the blind person would think of colors as what we call "gray scale". The analogical question is whether the components of intelligence that psychologists have posited are like colors in that they 'seem' as if they are 'unique' objects or is there a single number which we may obtain from the components.[8] Is this single number like colors or is it like the gray-scale?

7. COMPLEXITY AND HIERARCHY

The concept of layering or hierarchy is one of the most basic in the universe. Whereas hierarchy requires more detailed explanation the concept of layering is easier to envision and observed all over the world, at a very coarse-resolution. We use pictures of all sorts (as in Figure 2).

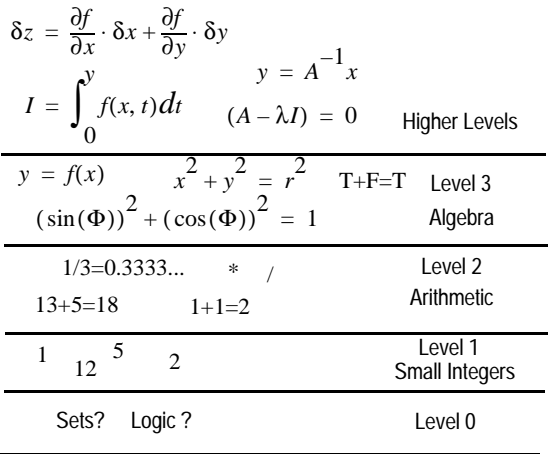


Figure 2: Highly-suggestive Layering in Mathematics: Knowledge is built-up in layers. New knowledge is built on top of old knowledge. This has significance for intelligence testing.

What better example than knowledge? *Data is raw. Information is data that is meaningful to an intelligent entity. Knowledge must be compressed information.* The only way to compress information is via exploiting regularities and pat-

terns. Since mathematics is the study patterns, and regularities of all kinds, it is clearly the best tool with which to do science. Many more examples of layering can be found [1],[5],[6].

Thus the *scientificity* (intensity) of knowledge must be mathematics. Is it possible to measure intelligence separate and apart from knowledge? Do we want to weight some kinds of knowledge more heavily than others?

8. DISTANCE & MEASUREMENT

The main problem here is whether, after having gone through the problem of identifying the various components of intelligence, we should multiply them or add them to create a single number called intelligence. Therefore two prototypical choices for distance are

$$d(\vec{x}, \vec{y}) = \left(\sum_{i=1}^n (\alpha_i x_i - \beta_i y_i)^{2m} \right)^{\frac{1}{2m}} \quad (3)$$

$$d(\vec{x}, \vec{y}) = \left[\prod_{i=1}^n x_i^{\alpha_i} \cdot y_i^{\beta_i} \right]^{\frac{1}{\Omega}} \quad (4)$$

Obviously, in Eq (4) every component must be nonzero. There are good reasons why it is so. If normal functioning of a human depends on having absolutely no genetic defects, and if the intelligence of a human is determined by n genes, then if any of them is defective it should effect the score in the same way that the reliability of a composite is the product of the reliabilities of its components. In this sense, then the factors are analogous to probabilities.

This is also how we humans apparently tend to evaluate intelligence, as can be seen in the schizoid labeling of the condition known as *idiot-savant*. Being apparently superhuman in one aspect of intellectual activity is not sufficient to escape the label ‘idiot’. It is said that *an expert knows everything about nothing whereas a generalist knows nothing about everything*. In an extension of this, then, today’s experts (i.e. engineers) are idiot-savants. Their social IQ is said to be low. Programs like Maple, then, are also idiot-savants.

9. AVERAGE-IZATION

Consider the problem of being a juror in a beauty pageant. We will be forced to use a kind of scale in Eq. (5) (below)

$$B(\vec{x}) = 1 - \left(\prod_{j=1}^n \left[\{x_j - \mu_j\}^{\alpha_j} \right]^{\frac{1}{\Omega}} \right) \quad (5)$$

where the μ_i are the means. For example, the features/properties (of the vector \vec{x}) may be nose length, skin color, lip thickness, fatness, etc. We will not want to vote for those with lips too thin or too thick, with noses that are too long, or too short, legs too thin or too thick, skin too pale or too dark. In other words, we are not looking for the minimum or the maximum but rather the most perfect average there is (with some caveats). This is a different kind of logic, triage logic [10].

Then, the human-kind of intelligence, if it is going to resemble what we humans normally think about perfection (apparently) should be measured via

$$I(\vec{x}) = 1 - \left[\prod_{i=1}^n \{x_i - \mu_i\}^{\alpha_i} \right]^{\frac{1}{\Omega}} \quad (6)$$

where the $\{x\}$ are the various attributes of intelligence. The Turing test is probably for this kind of intelligence. For example, a machine that can solve differential equations and multiply 20 by 20 matrices in a jiffy (such as Maple, a Computer Algebra System) would flunk the Turing test. A human would know that a normal human (or maybe even an abnormal human) cannot do that. Therefore, the machine that could pass the Turing test would either have to be designed dumbed-down or it would have to learn to deceive. There are other things machines can do very quickly that humans cannot accomplish.

Thus the ‘measure’ above would show that such an entity could not be human (*ceteris paribus*, of course). In other words, as long as the machine is able to do the other things more or less as a human, then overachieving (outdoing humans) in one of the dimensions of the vector space would mark it as a machine.

Exactly the same would apply in some other capability such as being able to lift a few tons, swimming or running at superhuman speeds etc. For machines, then locomotion, would also be treated as part of intelligence. However, since even lower animals (less intelligent than us) can move around, it should not contribute much to the measurement of intelligence.

There are some psychologists who want to include many human capabilities, such as physico-kinetic intelligence (i.e. physical ability) in the intelligence equation. Therefore, this ‘autonomy’ capability of animals/machines may also be considered to be a part of intelligence. We may take those that have been posited by psychologists as a starting point keeping in mind that some of them may really be substitutes for each other so that the measurement might be more complicated.

10. MORE SOPHISTICATION

Consider the simple problem of nutrition. Suppose we can create a balanced diet from the few foods available from three separate food groups; meat (protein), carbohydrates, and vegetables as shown below.

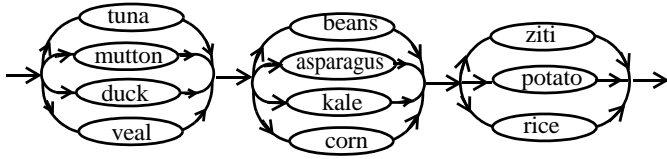


Figure 3: Parallel or Serial Choices. The problem is actually about multiplication vs addition. Diagrams such as this occur in electrical circuits, Boolean circuits [9], or choice making.

In terms of circuit analysis (which can be thought of in terms of Boolean algebra, [9]) it is clear that the parallel lines are about choices (and thus lack of constraints) and therefore represent logical-OR (disjunction), whereas the seriality/sequentiality denotes a logical-AND (conjunction). Probably the first thing a statistician would do if faced with the problem of determining the relationship between food groups and a balanced diet would be to try correlation-regression analysis which would be nothing more than

$$N = \alpha_0 + \alpha_1 t + \alpha_2 m + \dots + \alpha_n c \quad (7)$$

where t =tuna, m =mutton, c =corn etc. This is really the same kind of valuation of the problem as a weighted average. However, if we think logically then we should be considering a function of form;

$$N = MCV \quad (8)$$

since we need to ingest food from all the groups. Furthermore, since these food groups may be instantiated via specific examples, then using fuzzy logic, we should be regressing one of

$$N = (t + m + d + v)(b + a + k + c)(z + p + r) \quad (9a)$$

$$N = (t + m + d + v)^\alpha (b + a + k + c)^\beta (z + p + r)^\gamma \quad (9b)$$

Obviously, the latter form (Eq. 9) is not only correct but will result in many products (possibly to various powers). It is exactly this kind of products that dimensional analysis produces however it works only for problems with physical dimensions. However, there are methods that will produce similar equations for any problem if sufficient amount of data is available [4]. If intelligence-measurement is at least as complex as that of proper nutrition, then the simple weighted average kind of methods which are additive will not work. In other words the regression in Eq (7) is something like a combination of logical (or fuzzy) ORs and ANDs. A question that comes to mind is if there are fuzzy operators which are neither OR nor AND but something like both and exactly like neither. The special functions [11]

$$H_h(x, y) = \frac{1}{2} \cdot (x + y)^{h+1} \quad (10a)$$

$$M_m(x, y) = 2^m \cdot \left(\frac{(x-y)^2}{2[(x-y)^2]^{1/2}} \right)^{m+1} \quad (10b)$$

or others similar to these can be used in cases in which we are not sure if additive or multiplicative models should be used. One can show that [11]

$$Max(x, y) = H_o(x, y) + M_o(x, y) \quad (11a)$$

$$Min(x, y) = H_o(x, y) - M_o(x, y) \quad (11b)$$

Therefore the operator (fuzzy t -co-norm)

$$F(x, y) = H_o(x, y) + (2\xi - 1)M_o(x, y) \quad (11c)$$

is neither a norm (intersection) or conorm (union) but a fuzzy operator or a fuzzy norm since it is a norm for $\xi = 0$ and a conorm for $\xi = 1$. Some of the present day attributes of intelligence posited by psychologists probably are substitutes for each other and thus Eq (6) might distort the measurement. Therefore, something like Eq (9) where the additions are fuzzy unions and fuzzy intersections will probably give better results. The equations are readily and intuitively comprehensible in terms of theory of reliability based on probability. Fuzzification of the norm-conorm can be done for any fuzzy logic. For example, the simple product/sum logic given by

$$i(x, y) = xy \quad (12a)$$

$$u(x, y) = x + y - xy \quad (12b)$$

can be easily fuzzified via

$$F(x, y) = \rho xy + (1 - \rho)(x + y - xy) \quad (12c)$$

11. HUMAN INTELLIGENCE

The main problem today in human intelligence tests (and genetics) is calculating how much of intelligence is ‘inherited’ and how much of it is learned. There are several ways in which the model for this may be derived. One way would be to point out general conditions which the ‘intelligence function’ must satisfy. It should be multiplicative. It should display the increase of intelligence in time from the time of birth. It should converge on some limit on average for the people while being

allowed to fluctuate about the average rate of increase and the limit of human intelligence. The equation

$$\frac{dx}{dt} = \lambda(\alpha - x) \quad (13)$$

increases exponentially, and converges to a limit which is a good approximation. We need to know what the parameters mean, and this can be gleaned from the behavior of the solution. In Fig (4a) we see several trajectories. Some converge to above average intelligence, and some to less than average. Obviously the coefficient α determines this limit.

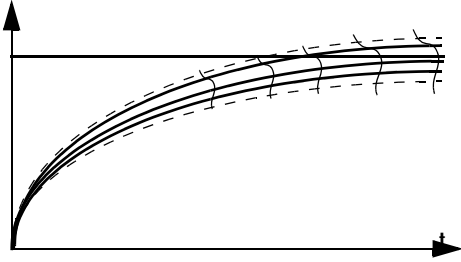


Figure 4a : Variations in α of the Intelligence Model .

In Fig (4b) we see a fluctuation in the rate of increase of intelligence, and this is controlled by the coefficient λ .

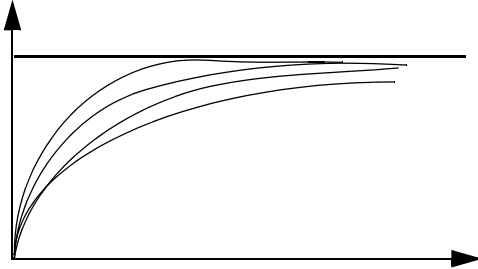


Figure 4b: Fluctuations in λ of the Intelligence Model

Logically both of these parameters then should be a function of both genetics and environment. Since we have determined that multiplicativity is important, the model should be

$$\frac{d}{dt}I(t) + \lambda G^{\eta} E^{\epsilon}(t) I(t) = \lambda \alpha G^{h+\eta} E^{e+\epsilon}(t) \quad (14)$$

Integrating it once and rearranging terms we obtain the integral equation

$$I(t) = K(t) - \lambda G^{\eta} \int_0^t E^{\epsilon}(\sigma) I(\sigma) d\sigma \quad (15a)$$

$$\text{with } K(t) = \alpha \lambda G^{h+\eta} \int_0^t E^{e+\epsilon}(s) ds \quad (15b)$$

which is exactly what most researchers claim, that is, intelligence at time t , that is $I(t)$, is a function of the past interaction of intelligence with environment summed up over time from time zero (birth) to the present time t . The interaction is multiplicative as it should be, and the equation is a reasonably good approximation over time of how living things (especially humans) learn. The solution is

$$I(t) = \Gamma e^{\lambda G^{\eta} \int_0^t E^{\epsilon}(\tau) d\tau} \int_0^t E^{e+\epsilon}(s) e^{-\lambda G^{\eta} \int_s^t E^{\epsilon}(\tau) d\tau} ds \quad (16)$$

where $\Gamma = \lambda \alpha G^{h+\eta}$ which in the limit goes to

$$I = \alpha E^e G^h \quad (17)$$

If one day robots which learn from their environment are created, similar equations will be good first order approximations. Same probability techniques can be used on these equations, and statistics such as 'heritability' can be calculated. If the multiplication above is treated as some kind of a fuzzy intersection, then we can see quite clearly that the same kind of an equation can easily 'explain' the existence of natural language among living things. At the limits the equation must reduce the crisp logic, and we can see that it does. Only in the case when both genetic capability is there and when there is proper environmental stimulation, does language exist. If one or the other is missing there is no language. We can show how this equation explains what psychologists have said (in words) for a long time. Computing the virtual variation, we obtain for the special (and simpler case) of $e = h = \alpha = 1$

$$dI = E \cdot dH + H \cdot dE \quad (18)$$

If the environment is enriched, the corresponding increase in intelligence depends on the genetic capability. Thus putting a dog in school cannot give it human level intelligence. Similarly, if there is a change in the genetic make-up (e.g. the difference between a chimp and a human) the change in the intelligence depends on the environment. A human brought up without human contact cannot walk or talk or dress up.

APPENDIX

Exact Differentials and Path Functions

The distinction between the related concepts *state* and *process* is an important one. There are mathematical definitions and consequences of these ideas. A *state* (or property) is a *point function*. The state of any system is the values of its *state vector* (a bundle of properties which characterizes a system). If we use these variables as coordinates then any state of the system is a point in this n-dimensional space of properties/characteristics. Conversely each state of the system can be represented by a single point on the diagram (of this space). For example for an ideal gas the state variables are temperature, pressure, volume, etc. Each color can be represented as a point in the 3-D space spanned by the R, G and B vectors. Intelligence is commonly accepted to be a state variable, i.e. a point. The scalar, Spearman's g , (single number, not a vector) can be obtained from this vector by using a distance metric. The argument that the values of the components cannot be obtained from the scalar, g , may be valid depending on the distance metric however, the distance metric may be devised in a way in which the components can be obtained from the scalar. Distance on a metric space is a function only of the end points i.e. between two states. However, the determination of some quantities requires more than the knowledge simply of the end states but requires a specification of a particular path between these points. These are called *path functions*. The commonest example of a path function is the length of a curve. Another example is the work done by an expanding gas. So is Q , the heat (transferred). In that sense work and heat are interactions between systems (i.e. processes), not characteristics of systems (i.e. state parameters/variables). Intuitively, when we talk about small changes or small quantities we use the differentials dx or δx . However the crucial difference is that although there may exist a function such that

$$\int_b^a dF = \int_b^a f(x)dx = F(x)\Big|_b^a = F(a) - F(b) \quad (A.1)$$

there is no function Q , (heat) such that

$$\int_b^a \delta q = Q(x)\Big|_b^a = Q(a) - Q(b) \quad (A2)$$

Instead we write

$$\int_a^b \delta q = Q_{ab} \quad (A3)$$

meaning that Q_{ab} is the quantity of heat transferred during the process from point a to point b . Similarly because the infinitesimal length of a curve in the plane is given by

$$ds = \sqrt{dy^2 + dx^2} \quad (A4)$$

we cannot integrate ds to obtain

$$S(b) - S(a) = \int_a^b \delta s = \int_a^b ds \quad (A5)$$

but instead first the curve $y=f(x)$ must be specified. Equivalently, if z is a function of two independent variables x and y , and this relationship is given by $z=f(x,y)$ then z is a point function. The differential dz of a point function is an *exact differential* and given by

$$dz = \left(\frac{\partial z}{\partial x}\right)dx + \left(\frac{\partial z}{\partial y}\right)dy \quad (A6)$$

Consequently if a differential of form $dz = Mdx + Ndy$ is given, it is an exact differential only if

$$\frac{\partial M}{\partial y} = \frac{\partial N}{\partial x} \quad (A7)$$

Therefore in the mathematical function used for the simple two-factor (nature-nurture) *Intelligence Function* the *environmental path taken* does make a difference in the final result which is assumed to be a state function (although computed from mental processes).

REFERENCES

1. Hubey, H.M. (1996) Topology of Thought, CC-AI: The Journal for the Integrated Study of Artificial Intelligence, Cognitive Science, and Applied Epistemology, vol 13, No.2-3, 225-292.
2. Eccles, J. (1989) Evolution of the Brain: Creation of the Self, Routledge, New York
3. Schmidt-Nielsen, K. (1984) Scaling: Why is Animal Size So Important?, University of Cambridge Press, Cambridge.
4. Hubey, H.M. (2000) A Complete Unified Method for Taming the Curse of Dimensionality in Datamining and Allowing Logical-ANDs in ANNs, submitted to Data-mining and Knowledge Discovery.
5. Hubey, H.M. (1994) Mathematical and Computational Linguistics, Mir Domu Tvoemu, Moscow, Russia.
6. NIST (2000) White Paper
7. Hubey, H.M. (2001) Evolution of Intelligence:Direct Modeling of Temporal Effects of Environment on a Global Absolute Scale vs Statistics, accepted by Kybernetes: The International Journal of Systems and Cybernetics.
8. Hubey, H.M. (1996) Logic, physics, physiology, and topology of color, Behavioral and Brain Sciences, vol 20:2, pp.191-194.
9. Hubey, H.M. (1999) Mathematical Foundations of Linguistics, Lincom Europa, Muenchen, Germany.
10. Hubey, H.M. (2000b) Fuzzy Logic and Calculus of Beauty, Moderation, and Triage, The Proceedings of the 2000 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS2000), June 26-29, Las Vegas.
11. Hubey, H.M. (1999) The Diagonal Infinity, World Scientific, Singapore.

Evaluation of System Intelligence with Pictorial Data Visualization

V. Grishin*, A. Meystel[°]

* View Trends, Ltd., Cleveland, OH

[°] Drexel University, Philadelphia, PA

Extended Abstract

1. Introduction

Aspects of the research. The concept of evaluating the intelligence of systems presented in this paper is based upon the model of intelligence outlined in [1] and the advancements in visualization described in [2]. Since the main mechanism of intelligence is the mechanism of generalization, it would be prudent to judge the degree of intelligence by the ability of the system to generalize. This ability can be detected by the means of visualization. Visualization of the system and/or the situation allows us to use the primary orientation of our visual capabilities to the situations and/or modes of functioning based upon “gestalt” i.e. capabilities to form a harmonious and consistent entity out of details.

We will explore the unique ability of the visualization systems to diagnose the system and/or its state by discovering the *syndrome*: a group of symptoms, or diagnostic features that collectively indicate or characterize a disease, a psychological disorder, or another abnormal condition which has some unity within itself. We will use the term syndrome for technological cases either to characterize some psychological disorder based upon an intrinsic or other unity. For example, a multiplicity of unfortunately coinciding factors can lead to a catastrophe. Thus, for this particular catastrophe, the combination of these factors is a *syndrome*.

Thus, our approach is pursuing two major goals. First, our intention is to solve the unsolved yet a fairly complicated problem of data mining and interpretation. This is a central problem of intelligence functioning: *how knowledge can be extracted from raw data* via visualization. Solving this problem would require analysis of the real world situations and constructing their models by effectively combining formal, verbal and even non-verbalized models of analyzed knowledge. It turns out that visualization can help the human decision maker to associate these diversified models and to formalize the new knowledge for the subsequent use in both manned and unmanned intelligent systems.

To accomplish this goal a human-computer dialog has to be constructed at each step of visualization. This dialog is aimed into restoration and analysis of the *hierarchy of features and descriptions* for states, situations, and scenes. It should provide for fast and accurate discrimination, description and understanding of *known and new* situation and their reasons. A visual-verbal language is created as a part of this dialog for each case of analysis individually. This approach provides effective selection of new models for discovered singularities, observed changes of situation, detected structures, etc.

An effective *interpretation* of visual-verbal results in terms of data properties and known models is the result of this approach. Although the human participant is a must, and there is no automated procedures to rely upon (as follows from its goals), a number of important advantages can be registered in comparison with automated systems neural networks, pattern recognition, etc. To get the good interpretation we use the *simple data mapping into pictures* and the human natural “gestalt-skills” for determining entities in these pictures.

Principles of the Human-Computer Dialog for Picture Analysis. The following main components of our *dialog realization* distinguish our approach from others by more effective use of human cognition:

1. *Constructing holistic images of exhaustively represented data about situation.* If the number of variables in a situation is more than 20-30, the matrix $N \times M$ is to be analyzed where each row displays a time series of one variables or each cell represents a current value of separate variables or others. A variable value is mapped into color-brightness. The ability to simultaneously

represent more than 1000*1000 numbers and to see some “general image” of situation is considered to be an advantage of our realization. This starting image allows for understanding a holistic structure of the situation, for detection some of the available skeletons, and for mapping its different properties into levels of some hierarchical organization that will direct the subsequent informative feature search.

2 *Combinatorial Searching in the Matrix* by permutations of rows, columns and cells. For each permutation the following is performed: sharpening the edges where required, smoothing where possible, value-to-color mapping adjustment where beneficial, etc. to get more informative, more interpretable image or at least to improve of its quality.

3. *Mapping from matrix to entities: individual patches, group patches, clusters of groups.* Informative variables and features found as a result of matrices permutation can be visualized by grouping the elementary units of image together. Thus, only tens of variables will be displayed at the levels of lower resolution but with complete mapping of their relationships within the image. This allows to determine shapes and more general forms that are more effective for visual analysis than color variation in the primary high resolution crowd of elementary patches. The dialog with generalized levels consists of a searching group variables and relationships among them.

4. *Selecting appropriate criteria of decisions.* It is important to underline that if no prior knowledge and/or hypotheses exist, then forming a syndrome is done based upon human gestalt skills and experiences. If there is some knowledge of situation and its evaluation criteria, the process of will synthesize this knowledge with the gestalt intuitions.

Pattern Analysis by Human Vision: Can It Be Automated? Human vision has an ability to quickly and efficiently compare several images in parallel with hundreds of local attributes and features related to the shapes, textures, colors, or brightness of the images. A standalone spot on a picture or a spot cluster has certain boundaries shapes which can be segmented using simple local features such as “straight line”, “concave”, “convex”, “angle”, “hole”, etc. These features have attributes as “sizes”, “orientation”, “symmetries”, etc. and are connected by means “upper-lower”, “left-right”, “inside-outside” adjacency. Local features are visually unified into more complicated shapes as “wave”, “leaf”, “a face profile” and others associated with real world objects and also have above mentioned and more complicated attributes.

This process of feature generalization continues up to holistic image of a spot including also its integral features as “complexity”, “symmetries”, “elongations” besides usual sizes, orientations, and position on the picture and others. In addition, the color and textural properties can be described. These attributes and features are then organized into a multilevel (multiresolutional) hierarchy that can be partially verbalized, or at least, tagged with symbols. If a picture contains many different separate shapes, such hierarchy can be constructed for these shapes clusters and clusters groups up to all picture. In addition, the combinatorial and statistical features could be visually detected and estimated. Vision rapidly moves through this hierarchy, searching for more details or generalizing the attributes allows for the simultaneous examination of many facets of the image by means of a variety of attributes and features.

Automated Description of Visual Patterns. Combinations of disjunctions and conjunctions of features Q_i and their attributes A_i can be applied for formalizing human representation of patterns. So called conjunctive normal form (CNF) describes some pattern with variation of features attributes, e.g. [“middle size”[Q_a **AND** Q_b [which is “symmetrical” around axis A_c] **AND** Q_k [known to be “small concave” and located in A_i (k) (e.g. position)]. Disjunctive normal forms ($DNF = CNF_1$ **OR** ($Q_g \vee A_k$) **OR** CNF_N) from separate features or complicated patterns describe picture classes with supplemental patterns. These descriptions are invariant for global rotations, shifts and projective transformations of whole shapes as well as their parts (with some limits). Similar formal tools can be applied with the purpose to formalize many other elements of the human-computer dialog. The transfer of knowledge from a human to a computer can be performed by using a subsystem of learning.

The following components of our research should be outlined:

2. Analysis of theoretical and experimental fundamentals that suggests that automated visualization is efficient in discovering entities, syndromes, and singularities.

Phase 1. Development of Automated Visualization System for Decision Making.

Usually data visualization is a *human-computer dialog* with the following general structure :

Stage 1. Entering data into the system and their consecutive processing in subsystems 1-4

Subsystem 1. Data gathering, transformation, filtration.

Subsystem 2. Mapping Data into visual paradigm, e.g. pictures.

Subsystem 3. Computer supported human visual analysis of the visualized data: features selection and transformation of situations into a visual relational map.

Subsystem 4. Comparison with a priori knowledge related to the features and the multi-feature formations and search of new ones.

Stage 2. Change of the chosen set of variables and parameters for the analysis and repetition of the cycle Of consecutive running of Subsystems 1-4.

Stage 3. Estimation of results, hypothesizing entities (syndromes), testing it through Subsystems 1-4 again, formalization and decision-making.

These three stages are run presently as a human-computer dialog that can have cycles between these stages in any order assigned by a human. We intend to automate this process by equipping the human-computer dialog processes by learning subsystem.

The strategy and the techniques of implementing the subsystem of learning and subsequent conducting the interpretation of results will be determined by the following factors:

- *goals are pursued within a particular domain and assignment*
- *limitations of combining human and computer capabilities*
- *available algorithms of generalization and instantiation*
- *metrics accepted for evaluating the performance and intelligence of the system.*

Phase 2. Application of Automated Visualization System for evaluating performance and Intelligence of Intelligent Systems.

In this case, the result of learning from the human during the human-computer dialog will be used for both: a) automatic analysis of data and b) for evaluating the performance and intelligence of intelligent systems.

Assume, an intelligent computer vision system has performed image processing. As a result of this, a particular image underwent a multiple generalization and the results of this are presented as the result of image analysis and interpretation. Let us consider another case: an intelligent system has planned a motion trajectory for an unmanned vehicle. In order to evaluate the intelligence of these systems, their problem solutions are presented to the automated system of visualization. The structure of the image and the structure of the motion trajectory are visualized and the prospective syndromes are obtained. The results of visualization are compared with the results of processing by the system undergoing testing. This comparison serves as the estimate of performance and intelligence.

Visualization can be used not for states but for the state-space trajectories. It seems natural to expand the process of visualization from evaluation of states and situations to evaluation of state space trajectories as a whole. This would allow for comparison of different system behaviors by means of visualization of appropriate data. The results of visualization in this case are not the *images*, or *pictures* but rather *movies*. There is plenty of evidence that the gestalt abilities can be applied not only to static images but also to their consecutive strings that represent *processes*. Finding a temporal unity of a process is the problem that has never be proposed before as a problem for the system of automatic visualization.

Intelligence is defined as a faculty of a system that increases the probability of successful functioning in a variety of problem solving situations and under uncertainty of the conditions of the environment.

When systems function, the results of their functioning reflect not only changes of the environments and the goals assigned but also the results of their control system generating decisions and shaping processes. The consistency of control system functioning will be reflected in a temporal gestalt of processes that are generated as a result of control. It is our hypothesis that one can judge the control system by observing the output and not only measuring how close it is to the output specifications but also how *satisfactorily* the system responds to all changes. Since, the construction of a metric that evaluates responses to all changes is a problematic one (H-infinity is one of the efforts) and since the combination of uncertain circumstances has unlimited number of possible combinations, we assume that using the natural ability of human vision to register and recognize *singularities* of external images, the ability to distinguish differences in response can be detected via visualization.

3. Existing Experience of Using Visualization for the Purposes of Recognizing Singularities in Functioning Systems Confirms Our Hypotheses.

Our experiences in visualization system development for human decision making support has shown that appropriate *data visualization* can :

- drastically enhance efficiency in comparing different approaches of intelligence,
- specify the most effective field of each approach application and combine many of them to built an intelligent system for wide diversity of environment variations and control tasks (or whatsoever...),
- extend this system capabilities for some set of important but uncertain (unpredictable) situations by means their holistic visualization and recognition in real time.

Gas-turbine engine diagnostics in airplanes and search for the cardiology diagnostic syndrome demonstrate capabilities of visualization techniques (see Figure 1 and 2). Analysis of existing experimental data allowing to expect that the proposed method of intelligence evaluation can be successful. Pictorial visualization has allowed to analyze the *transition* modes of engines and temporal processes of human heart functioning. As a result, the effect of much earlier symptoms of many malfunctions in the *transition* modes of operation were discovered to be different from the static modes, and more reliability of conclusions was achieved.

Interpretation of the successful use of visualization. The following factors were taken in account:

- What was special in the way we have arranged the process of visualization
- What does it suggest for the future organization of visualization
- The “Hypothesis of Visualization” that we have arrived at
- Introduction of the concepts: temporal gestalt, dynamic syndrome, visualization of transition modes.

The recommended use of visualization for intelligence testing include:

- The specifics of intelligence testing
- The similarities of the case of intelligence testing and examples
- Restatement of the Hypothesis of visualization for the case of intelligence evaluation.
- How it will be applied for the cases of
 - planner/controller for industrial crane
 - autonomous unmanned vehicle

References:

1. A. Meystel, “Evolution of Intelligent System Architectures: What Should be Measured?”, Proceedings of the NIST Workshop on Metrics for performance and Intelligence of Intelligent Systems,” Gaithersburg, MD, 2000
2. V. Grishin, “Multivariate Data Visualization for Qualitative Model Choice in Learning System,” in Proceedings of the 1998 IEEE International Symposium on Intelligent Control, NIST, Gaithersburg, 1998, pp. 622-627

